



Reltool

Copyright © 2009-2024 Ericsson AB, All Rights Reserved
Reltool 0.9.1.1
December 5, 2024

Copyright © 2009-2024 Ericsson AB, All Rights Reserved

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. The Initial Developer of the Original Code is Ericsson AB. Ericsson AB, All Rights Reserved.

December 5, 2024

1 Reltool Users Guide

Reltool is a release management tool. It analyses a given Erlang/OTP installation and determines various dependencies between applications. The graphical frontend depicts the dependencies and enables interactive customization of a target system. The backend provides a batch interface for generation of customized target systems.

1.1 Introduction

Reltool is a release management tool. It analyses a given Erlang/OTP installation and determines various dependencies between applications. The graphical frontend depicts the dependencies and enables interactive customization of a target system. The backend provides a batch interface for generation of customized target systems.

1.1.1 Scope and Purpose

This manual describes the Reltool application, as a component of the Erlang/Open Telecom Platform development environment. It is assumed that the reader is familiar with the Erlang Development Environment, which is described in a separate User's Guide.

1.1.2 Prerequisites

The following prerequisites is required for understanding the material in the Reltool User's Guide:

- familiarity with Erlang/OTP system principles and Erlang/OTP design principles

The application requires Erlang/OTP release R13B02 or later.

1.1.3 About This Manual

In addition to this introductory chapter, the Reltool User's Guide contains the following chapters:

- Chapter 2: "Usage" describes the architecture and typical usage of the application.
- Chapter 3: "Examples" gives some usage examples

1.1.4 Where to Find More Information

Refer to the following documentation for more information about Reltool and about the Erlang/OTP development system:

- the Reference Manual of Reltool
- the Erlang/OTP System Principles
- the Erlang/OTP Design Principles
- Programming Erlang: Software for a Concurrent World (2007), Pragmatic Bookshelf, ISBN13: 9781934356005.

1.2 Usage

1.2.1 Overview

This document focuses on the graphical parts of the tool. The concepts are explained in the reference manual for the module `reltool`.

1.3.2 Inspecting the configuration

```
Erlang/OTP 20 [erts-9.0] [source-c13b302] [64-bit] [smp:4:4] [ds:4:4:10] [async-threads:10]
[hipe] [kernel-poll:false]
Eshell V9.0 (abort with ^G)
1>
1> Config = {sys, [{escript, "examples/display_args", [{incl_cond, include}]},
  {app, inets, [{incl_cond, include}]},
  {app, mnesia, [{incl_cond, exclude}]},
  {app, ssl, [{incl_cond, exclude}]},
  {app, runtime_tools, [{incl_cond, exclude}]},
  {app, syntax_tools, [{incl_cond, exclude}]}]}.
{sys, [{escript, "examples/display_args", [{incl_cond, include}]},
  {app, inets, [{incl_cond, include}]},
  {app, mnesia, [{incl_cond, exclude}]},
  {app, ssl, [{incl_cond, exclude}]},
  {app, runtime_tools, [{incl_cond, exclude}]},
  {app, syntax_tools, [{incl_cond, exclude}]}]}
2>
2> {ok, Server} = reltool:start_server([Config]).
{ok,<0.66.0>}
3>
3> reltool:get_config(Server).
{ok,{sys,[{escript,"/usr/local/lib/erlang/lib/reltool-0.7.3/examples/display_args",
  [{incl_cond,include}]},
  {app,inets,[{incl_cond,include}]},
  {app,mnesia,[{incl_cond,exclude}]},
  {app,runtime_tools,[{incl_cond,exclude}]},
  {app,ssl,[{incl_cond,exclude}]},
  {app,syntax_tools,[{incl_cond,exclude}]}]}]}
4>
4> reltool:get_config(Server, false, false).
{ok,{sys,[{escript,"/usr/local/lib/erlang/lib/reltool-0.7.3/examples/display_args",
  [{incl_cond,include}]},
  {app,inets,[{incl_cond,include}]},
  {app,mnesia,[{incl_cond,exclude}]},
  {app,runtime_tools,[{incl_cond,exclude}]},
  {app,ssl,[{incl_cond,exclude}]},
  {app,syntax_tools,[{incl_cond,exclude}]}]}]}
5>
5> reltool:get_config(Server, true, false).
{ok,{sys,[{root_dir,"/usr/local/lib/erlang"},
  {lib_dirs,[]},
  {escript,"/usr/local/lib/erlang/lib/reltool-0.7.3/examples/display_args",
    [{incl_cond,include}]},
  {mod_cond,all},
  {incl_cond,derived},
  {app,inets,
    [{incl_cond,include},{vsn,undefined},{lib_dir,undefined}]},
  {app,mnesia,[{incl_cond,exclude}]},
  {app,runtime_tools,[{incl_cond,exclude}]},
  {app,ssl,[{incl_cond,exclude}]},
  {app,syntax_tools,[{incl_cond,exclude}]},
  {boot_rel,"start_clean"},
  {rel,"start_clean","1.0",[]},
  {rel,"start_sasl","1.0",[sasl]},
  {emu_name,"beam"},
  {relocatable,true},
  {profile,development},
  {incl_sys_filters,[".*"]},
  {excl_sys_filters,[]},
  {incl_app_filters,[".*"]},
  {excl_app_filters,[]}]},
```

```

    {incl_archive_filters,[".*"]},
    {excl_archive_filters,[...]|...}},
    {archive_opts,[]},
    {rel_app_type,...},
    {...}|...}}
6>
6> reltool:get_config(Server, true, true).
{ok,{sys,[{root_dir,"/usr/local/lib/erlang"},
  {lib_dirs,[]},
  {escript,"/usr/local/lib/erlang/lib/reltool-0.7.3/examples/display_args",
    [{incl_cond,include}]},
  {mod_cond,all},
  {incl_cond,derived},
  {erts,[{app,erts,
    [{vsn,"10.0"},
     {lib_dir,"/usr/local/lib/erlang/lib/erts-10.0"},
     {mod,erl_prim_loader,[]},
     {mod,erl_tracer,[]},
     {mod,erlang,[]},
     {mod,erts_code_purger,[]},
     {mod,erts_dirty_process_signal_handler,[]},
     {mod,erts_internal,[]},
     {mod,erts_literal_area_collector,[]},
     {mod,init,[]},
     {mod,erl_init,...},
     {mod,...},
     {...}|...}]},
  {app,compiler,
    [{vsn,"7.0.4"},
     {lib_dir,"/usr/local/lib/erlang/lib/compiler-7.0.4"},
     {mod,beam_a,[]},
     {mod,beam_asm,[]},
     {mod,beam_block,[]},
     {mod,beam_bs,[]},
     {mod,beam_bsm,[]},
     {mod,beam_clean,[]},
     {mod,beam_dead,[]},
     {mod,beam_dict,[]},
     {mod,beam_disasm,[]},
     {mod,beam_except,[]},
     {mod,beam_flatten,...},
     {mod,...},
     {...}|...}],
  {app,crypto,
    [{vsn,"3.7.4"},
     {lib_dir,"/usr/local/lib/erlang/lib/crypto-3.7.4"},
     {mod,crypto,[]},
     {mod,crypto_ec_curves,[]}]},
  {app,hipec,
    [{vsn,"3.15.4"},
     {lib_dir,"/usr/local/lib/erlang/lib/hipec-3.15.4"},
     {mod,cerl_cconv,[]},
     {mod,cerl_closurean,[]},
     {mod,cerl_hipeify,[]},
     {mod,cerl_lib,[]},
     {mod,cerl_messagean,[]},
     {mod,cerl_pmatch,[]},
     {mod,cerl_prettypr,[]},
     {mod,cerl_to_icode,[]},
     {mod,cerl_typean,...},
     {mod,...},
     {...}|...}],
  {app,inets,
    [{incl_cond,include},
     {vsn,"6.3.9"},

```

1.3 Examples

```
{lib_dir, "/usr/local/lib/erlang/lib/inets-6.3.9"},
{mod, ftp, []},
{mod, ftp_progress, []},
{mod, ftp_response, []},
{mod, ftp_sup, []},
{mod, http_chunk, []},
{mod, http_request, []},
{mod, http_response, ...},
{mod, ...},
{...}|...}},
{app, kernel,
 [{vsn, "5.2"},
  {lib_dir, "/usr/local/lib/erlang/lib/kernel-5.2"},
  {mod, application, []},
  {mod, application_controller, []},
  {mod, application_master, []},
  {mod, application_starter, []},
  {mod, auth, []},
  {mod, code, []},
  {mod, code_server, ...},
  {mod, ...},
  {...}|...}],
{app, mnesia, [{incl_cond, exclude}]},
{app, runtime_tools, [{incl_cond, exclude}]},
{app, sasl,
 [{vsn, "3.0.3"},
  {lib_dir, "/usr/local/lib/erlang/lib/sasl-3.0.3"},
  {mod, alarm_handler, []},
  {mod, erlrv, []},
  {mod, format_lib_sup, []},
  {mod, misc_sup, ...},
  {mod, ...},
  {...}|...}],
{app, ssl, [{incl_cond, exclude}]},
{app, stdlib,
 [{vsn, "3.3"},
  {lib_dir, "/usr/local/lib/erlang/lib/stdlib-3.3"},
  {mod, array, []},
  {mod, base64, ...},
  {mod, ...},
  {...}|...}],
{app, syntax_tools, [{incl_cond, exclude}]},
{app, tools,
 [{vsn, "2.9.1"}, {lib_dir, [...]}, {mod, ...}, {...}|...}],
{boot_rel, "start_clean"},
{rel, "start_clean", "1.0", []},
{rel, "start_sasl", "1.0", [...]},
{emu_name, "beam"},
{relocatable, true},
{profile, ...},
{...}|...}}

7>
7> reltool:get_config([sys, [{profile, embedded}]]), true, false).
{ok, {sys, [{root_dir, "/usr/local/lib/erlang"},
  {lib_dirs, []},
  {mod_cond, all},
  {incl_cond, derived},
  {boot_rel, "start_clean"},
  {rel, "start_clean", "1.0", []},
  {rel, "start_sasl", "1.0", [sasl]},
  {emu_name, "beam"},
  {relocatable, true},
  {profile, embedded},
  {incl_sys_filters, ["^bin", "^erts", "^lib", "^releases"]},
  {excl_sys_filters, ["^bin/(erlc|dialyzer|typer)(|\\.exe)$",
```

```

        "^erts.*bin/(erlc|dialyzer|typer)(|\\.exe)$",
        "^erts.*bin/.*(debug|pdb)" ]},
    {incl_app_filters, ["^ebin", "^include", "^priv"]},
    {excl_app_filters, []},
    {incl_archive_filters, [".*"]},
    {excl_archive_filters, ["^include$", "^priv$"]},
    {archive_opts, []},
    {rel_app_type, permanent},
    {embedded_app_type, load},
    {app_file, keep},
    {debug_info, keep}}}]
8>
8> reltool:get_config([sys, [{profile, standalone}]]], true, false).
{ok, {sys, [{root_dir, "/usr/local/lib/erlang"},
    {lib_dirs, []},
    {mod_cond, all},
    {incl_cond, derived},
    {boot_rel, "start_clean"},
    {rel, "start_clean", "1.0", []},
    {rel, "start_sasl", "1.0", [sasl]},
    {emu_name, "beam"},
    {relocatable, true},
    {profile, standalone},
    {incl_sys_filters, ["^bin/(erl|epmd)(|\\.exe|\\.ini)$",
        "^bin/start(|_clean).boot$", "^erts.*bin", "^lib$"]},
    {excl_sys_filters, ["^erts.*bin/(erlc|dialyzer|typer)(|\\.exe)$",
        "^erts.*bin/(start|escript|to_erl|run_erl)(|\\.exe)$",
        "^erts.*bin/.*(debug|pdb)" ]},
    {incl_app_filters, ["^ebin", "^priv"]},
    {excl_app_filters, ["^ebin/.*\\.appup$"]},
    {incl_archive_filters, [".*"]},
    {excl_archive_filters, ["^include$", "^priv$"]},
    {archive_opts, []},
    {rel_app_type, permanent},
    {app_file, keep},
    {debug_info, keep}}}]}
```


ok


```

    [{copy_file,"driver_int.h"},
     {copy_file,"ei_connect.h"},
     {copy_file,"ei.h"},
     {copy_file,"erl_nif_api_funcs.h"},
     {copy_file,"erl_fixed_size_int_types.h"},
     {copy_file,"erl_int_sizes_config.h"},
     {copy_file,"erl_interface.h"},
     {copy_file,"eicode.h"},
     {copy_file,"erl_driver.h"},
     {copy_file,"erlang.idl"},
     {copy_file,[...]},
     {copy_file,...},
     {...}]]],
{create_dir,"erts-10.0",
 [{create_dir,"bin",
  [{copy_file,"start"},
   {copy_file,"ct_run"},
   {copy_file,"erlexec"},
   {copy_file,"dialyzer"},
   {copy_file,"beam.smp"},
   {copy_file,"run_erl"},
   {copy_file,"erl","erts-10.0/bin/dyn_erl"},
   {copy_file,"to_erl"},
   {copy_file,"epmd"},
   {copy_file,"erl_child_setup"},
   {copy_file,"heart"},
   {copy_file,[...]},
   {copy_file,...},
   {...}|...}],
 {create_dir,"lib",
  [{create_dir,"internal",
   [{copy_file,"liberts_internal.a"},
    {copy_file,"liberts_internal_r.a"},
    {copy_file,"libethread.a"},
    {copy_file,"README"}]],
  ]},
 {create_dir,"src",[{copy_file,"setuid_socket_wrap.c"}]},
 {create_dir,"doc",[]},
 {create_dir,"man",[]},
 {create_dir,"include",
  [{create_dir,"internal",
   [{create_dir,"i386",[{...}|...]},
    {copy_file,"erl_errno.h"},
    {copy_file,[...]},
    {copy_file,...},
    {...}|...]},
   {copy_file,"driver_int.h"},
   {copy_file,"erl_nif_api_funcs.h"},
   {copy_file,"erl_fixed_size_int_types.h"},
   {copy_file,"erl_int_sizes_config.h"},
   {copy_file,[...]},
   {copy_file,...},
   {...}]]]},
 {create_dir,"lib",
  [{archive,"compiler-7.0.4.ez",[]},
   [{create_dir,"compiler-7.0.4",
    [{create_dir,"src",
     [{copy_file,"beam_flatten.erl"},
      {copy_file,[...]},
      {copy_file,...},
      {...}|...]},
     {create_dir,"ebin",
      [{copy_file,[...]},{copy_file,...},{...}|...}]]}],
   {archive,"crypto-3.7.4.ez",[]},
   [{create_dir,"crypto-3.7.4",

```


2 Reference Manual

Reltool is a release management tool. It analyses a given Erlang/OTP installation and determines various dependencies between applications. The graphical frontend depicts the dependencies and enables interactive customization of a target system. The backend provides a batch interface for generation of customized target systems.

`app_file`

The value of this parameter overrides the parameter with the same name on system level.

`debug_info`

The value of this parameter overrides the parameter with the same name on system level.

`incl_app_filters`

The value of this parameter overrides the parameter with the same name on system level.

`excl_app_filters`

The value of this parameter overrides the parameter with the same name on system level.

`incl_archive_filters`

The value of this parameter overrides the parameter with the same name on system level.

`excl_archive_filters`

The value of this parameter overrides the parameter with the same name on system level.

`archive_opts`

The value of this parameter overrides the parameter with the same name on system level.

On module (mod) level, the following options are supported:

`incl_cond`

This parameter controls whether the module is included or not. By default the `mod_cond` parameter on application and system level will be used to control whether the module is included or not. The value of `incl_cond` overrides the module inclusion policy. `include` implies that the module is included, while `exclude` implies that the module is not included. `derived` implies that the module is included if it is used by any other included module.

`debug_info`

The value of this parameter overrides the parameter with the same name on application level.

DATA TYPES

```

options()      = [option()]
option()       = {config, config() | file()}
               | {trap_exit, bool()}
               | {wx_debug, term()}

config()       = {sys, [sys()]}
sys()          = {root_dir, root_dir()}
               | {lib_dirs, [lib_dir()]}
               | {profile, profile()}
               | {erts, app()}
               | {escript, escript_file(), [escript()]}
               | {app, app_name(), [app()]}
               | {mod_cond, mod_cond()}
               | {incl_cond, incl_cond()}
               | {boot_rel, boot_rel()}
               | {rel, rel_name(), rel_vsn(), [rel_app()]}
               | {rel, rel_name(), rel_vsn(), [rel_app()], [rel_opt()]}
               | {relocatable, relocatable()}
               | {app_file, app_file()}
               | {debug_info, debug_info()}
               | {incl_sys_filters, incl_sys_filters()}
               | {excl_sys_filters, excl_sys_filters()}
               | {incl_app_filters, incl_app_filters()}
               | {excl_app_filters, excl_app_filters()}
               | {incl_archive_filters, incl_archive_filters()}
               | {excl_archive_filters, excl_archive_filters()}
               | {archive_opts, [archive_opt()]}

app()          = {vsn, app_vsn()}
               | {lib_dir, lib_dir()}
               | {mod, mod_name(), [mod()]}
               | {mod_cond, mod_cond()}
               | {incl_cond, incl_cond()}
               | {debug_info, debug_info()}
               | {app_file, app_file()}
               | {excl_lib, excl_lib()}
               | {incl_sys_filters, incl_sys_filters()}
               | {excl_sys_filters, excl_sys_filters()}
               | {incl_app_filters, incl_app_filters()}
               | {excl_app_filters, excl_app_filters()}
               | {incl_archive_filters, incl_archive_filters()}
               | {excl_archive_filters, excl_archive_filters()}
               | {archive_opts, [archive_opt()]}

mod()          = {incl_cond, incl_cond()}
               | {debug_info, debug_info()}

rel_app()      = app_name()
               | {app_name(), app_type()}
               | {app_name(), [incl_app()]}
               | {app_name(), app_type(), [incl_app()]}

rel_opt()      = {load_dot_erlang, boolean()}

app_name()     = atom()
app_type()     = permanent | transient | temporary | load | none
app_vsn()      = string()
archive_opt    = zip_create_opt()
boot_rel()     = rel_name()
app_file()     = keep | strip | all
debug_info()   = keep | strip
dir()          = string()
escript()      = {incl_cond, incl_cond()}
escript_file() = file()
excl_app_filters() = regexps()
excl_archive_filters() = regexps()
excl_lib()     = otp_root
excl_sys_filters() = regexps()

```



```
start(Options) -> {ok, WindowPid} | {error, Reason}
```

Types:

```
Options = options()  
WindowPid = window_pid()  
Reason = reason()
```

Start a main window process with options

```
start_link(Options) -> {ok, WindowPid} | {error, Reason}
```

Types:

```
Options = options()  
WindowPid = window_pid()  
Reason = reason()
```

Start a main window process with options. The process is linked.

```
start_server(Options) -> {ok, ServerPid} | {error, Reason}
```

Types:

```
Options = options()  
ServerPid = server_pid()  
Reason = reason()
```

Start a server process with options. The server process identity can be given as an argument to several other functions in the API.

```
stop(Pid) -> ok | {error, Reason}
```

Types:

```
Pid = server_pid() | window_pid()  
Reason = reason()
```

Stop a server or window process