IDA 9.0 IDAPython Changes and porting guide (DRAFT)

Introduction

This guide provides information about what has been changed in the IDAPython API between IDA 8.4 and 9.0.

The largest change due to the removal of two modules:
* ida_struct
* ida_enum

ida_struct

Removed functions:

The functions below 8.4 are removed those under 9.0 are altenatives.

NOTE: It is recommended to take a look at the implementation of the idc.* functions mentioned below.

8.4	9.0
add_struc	idc.add_struc
add_struc_member	idc.add_struc_member
del_struc	idc.del_struc
del_struc_member	idc.del_struc_member
del_struc_members	see [example](#del_struct_members)
dyn_member_ref_array	
expand_struc	idc.expand_struc
get_best_fit_member	see [example](#get_best_fit_member)
get_first_struc_idx	
get_innermost_member	see [example](#get_innermost_member)
get_last_struc_idx	
get_max_offset	
get_member	
get_member_by_fullname	
get_member_by_id	
get_member_by_name	
get_member_cmt	idc.get_member_cmt
get_member_fullname	see [example](#get_member_fullname)

get_member_id	idc.get_member_id
get_member_name	idc.get_member_name
get_member_size	idc.get_member_size
get_member_struc	
get_member_tinfo	
get_next_member_idx	
get_next_struc_idx	
get_or_guess_member_tinfo	
get_prev_member_idx	
get_prev_struc_idx	
get_sptr	
get_struc	
get_struc_by_idx	
get_struc_cmt	idc.get_struc_cmt
get_struc_first_offset	
get_struc_id	idc.get_struc_id
get_struc_idx	
get_struc_last_offset	
get_struc_name	idc.get_struc_name
get_struc_next_offset	
get_struc_prev_offset	
get_struc_qty	
get_struc_size	idc.get_struc_size
is_anonymous_member_name	
is_dummy_member_name	
is_member_id	idc.is_member_id
is_special_member	
is_union	idc.is_union
is_varmember	
is_varstr	
retrieve_member_info	
save_struc	
set_member_cmt	idc.set_member_cmt
set_member_name	idc.set_member_name
set_member_tinfo	
set_member_type	idc.set_member_type
set_struc_align	
set_struc_cmt	idc.set_struc_cmt
set_struc_hidden	
set_struc_idx	
set_struc_listed	
set_struc_name	idc.set_struc_name
stroff_as_size	
struct_field_visitor_t	
unsync_and_delete_struc	
visit_stroff_fields	
visit_stroff_udms	

Removed methods and members

member_t

* by_til
* eoff
* flag
* get_size
* get_soff
* has_ti
* has_union
* id
* is_baseclass
* is_destructor
* is_dupname
* props
* soff

* this
* thisown
* unimem

struct_t

* age
* from_til
* get_alignment
* get_last_member
* get_member
* has_union
* id
* is_choosable
* is_copyof
* is_frame
* is_ghost
* is_hidden
* is_mappedto
* is_synced
* is_union
* is_varstr
* like_union
* members
* memqty
* ordinal
* props
* set_alignment
* thisown

struct_field_visitor_t

* visit_field

udm_visitor_t

* visit_udm

ida_enum

Removed functions

The functions below 8.4 are removed those under 9.0 are altenatives.

8.4	9.0
add_enum	idc.add_enum
add_enum_member	idc.add_enum_member
del_enum	idc.del_enum
del_enum_member	idc.del_enum_member
for_all_enum_members	
get_bmask_cmt	idc.get_bmask_cmt
get_bmask_name	idc.get_bmask_name
get_enum	idc.get_enum
get_enum_cmt	idc.get_enum_cmt
get_enum_flag	idc.get_enum_flag
get_enum_idx	
get_enum_member	idc.get_enum_member
get_enum_member_bmask	idc.get_enum_member_bmask
get_enum_member_by_name	idc.get_enum_member_by_name
get_enum_member_cmt	idc.get_enum_member_cmt
get_enum_member_enum	idc.get_enum_member_enum
get_enum_member_name	idc.get_enum_member_name
get_enum_member_serial	
get_enum_member_value	idc.get_enum_member_value

get_enum_name	idc.get_enum_name	
get_enum_name2		
get_enum_qty		
get_enum_size	idc.get_enum_size	
get_enum_type_ordinal		
get_enum_width	idc.get_enum_width	
get_first_bmask	idc.get_first_bmask	
get_first_enum_member	idc.get_first_enum_member	
get_first_serial_enum_member		
get_last_bmask	idc.get_last_bmask	
get_last_enum_member	idc.get_last_enum_member	
get_last_serial_enum_member		
get_next_bmask	idc.get_next_bmask	
get_next_enum_member	idc.get_next_enum_member	
get_next_serial_enum_member		
get_prev_bmask	idc.get_prev_bmask	
get_prev_enum_member	idc.get_prev_enum_member	
get_prev_serial_enum_member		
getn_enum		
is_bf	idc.is_bf	
is_enum_fromtil		
is_enum_hidden		
is_ghost_enum		
is_one_bit_mask		
set_bmask_cmt	idc.set_bmask_cmt	
set_bmask_name	idc.set_bmask_name	
set_enum_bf	idc.set_enum_bf	
set_enum_cmt	idc.set_enum_cmt	
set_enum_flag	idc.set_enum_flag	
set_enum_fromtil		
set_enum_ghost		
set_enum_hidden		
set_enum_idx		
set_enum_member_cmt	idc.set_enum_member_cmt	
set_enum_member_name	idc.set_enum_member_name	
set_enum_name	idc.set_enum_name	
set_enum_type_ordinal		
set_enum_width	idc.set_enum_width	

enum_member_visitor_t

* visit_enum_member

ida_typeinf

Removed functions

* callregs_t_regcount
* get_ordinal_from_idb_type
* is_autosync
* get_udm_tid
* get_tinfo_tid
* tinfo_t_get_stock
* create_tinfo
* get_tinfo_details
* get_ordinal_qty

Added functions

* detach_tinfo_t(_this: "tinfo_t") -> "bool"
* get_tinfo_by_edm_name(tif: "tinfo_t", til: "til_t", mname: "char const *") -> "ssize_t"
* stroff_as_size(plen: "int", tif: "tinfo_t", value: "asize_t") -> "bool"
* visit_stroff_udms(sfv: "udm_visitor_t", path: "tid_t const *", disp: "adiff_t *", appzero: "bool") -> "adiff_t *"

* is_one_bit_mask(mask: "uval_t") -> "bool"
* get_idainfo_by_udm(flags: "flags64_t *", ti: "opinfo_t", set_lzero: "bool *", ap: "array_parameters_t", udm: "udm_t") -> "bool"

Added class

udm_visitor_t

* visit_udm

Removed methods

enum_type_data_t

* get_constant_group

Added methods

callregs_t

* set_registers(self, kind: "callregs_t::reg_kind_t", first_reg: "int", last_reg: "int") -> "void"

enum_type_data_t

* all_constants(self)
* all_groups(self, skip_trivial=False
* get_constant_group(self, *args) -> "PyObject *"
* get_max_serial(self, value: "uint64") -> "uchar"
* get_serial(self, index: "size_t") -> "uchar"

func_type_data_t.

* find_argument(self, *args) -> "ssize_t"

til_t

* find_base(self, n: "char const *") -> "til_t *"
* get_type_names(self) -> "const char *"

tinfo_t

* detach(self) -> "bool"
* is_punknown(self) -> "bool"
* get_enum_nmembers(self) -> "size_t"
* is_empty_enum(self) -> "bool"
* get_enum_width(self) -> "int"
* calc_enum_mask(self) -> "uint64"
* get_edm_tid(self, idx: "size_t") -> "tid_t"
* is_udm_by_til(self, idx: "size_t") -> "bool"
* set_udm_by_til(self, idx: "size_t", on: "bool"=True, etf_flags: "uint"=0) -> "tinfo_code_t"
* set_fixed_struct(self, on: "bool"=True) -> "tinfo_code_t"
* set_struct_size(self, new_size: "size_t") -> "tinfo_code_t"
* is_fixed_struct(self) -> "bool"
* get_func_frame(self, pfn: "func_t const *") -> "bool"
* is_frame(self) -> "bool"
* get_frame_func(self) -> "ea_t"
* set_enum_radix(self, radix: "int", sign: "bool", etf_flags: "uint"=0) -> "tinfo_code_t"
* rename_funcarg(self, index: "size_t", name: "char const *", etf_flags: "uint"=0) -> "tinfo_code_t"
* set_funcarg_type(self, index: "size_t", tif: "tinfo_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* set_func_rettype(self, tif: "tinfo_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* del_funcargs(self, idx1: "size_t", idx2: "size_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* del_funcarg(self, idx: "size_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* add_funcarg(self, farg: "funcarg_t", etf_flags: "uint"=0, idx: "ssize_t"=-1) -> "tinfo_code_t"
* set_func_cc(self, cc: "cm_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* set_funcarg_loc(self, index: "size_t", argloc: "argloc_t", etf_flags: "uint"=0) -> "tinfo_code_t"

* set_func_retloc(self, argloc: "argloc_t", etf_flags: "uint"=0) -> "tinfo_code_t"
* get_stkvar(self, insn: "insn_t const &", x: "op_t const", v: "sval_t") -> "ssize_t"

udm_t

* is_retaddr(self) -> "bool"
* is_savregs(self) -> "bool"
* is_special_member(self) -> "bool"
* is_by_til(self) -> "bool"
* set_retaddr(self, on: "bool"=True) -> "void"
* set_savregs(self, on: "bool"=True) -> "void
* set_by_til(self, on: "bool"=True) -> "void"

udtmembervec_t

* set_fixed(self, on: "bool"=True) -> "void"

Modified methods:

tinfo_t

8.4	9.0
find_udm(self, udm: "udmt_t *", strmem_flags: "int") -> "int"	find_udm(self, udm: "udmt_t *", strmem_flags: "int") -> "int"
	find_udm(self, name: "char const *", strmem_flags: "int") -> "int"
get_type_by_edm_name(self, mname: "const char *", til: "til_t"=None) -> "bool"	get_edm_by_name(self, mname: "char const *", til: "til_t"=None) -> "ssize_t"

ida_frame

8.4
To access the structure of a function frame, use:
* get_struc() (use func_t::frame as structure ID)
* get_frame(const func_t *pfn)
* get_frame(ea_t ea)

9.0
To access the structure of a function frame, use:
* tinfo_t::get_func_frame(const func_t *pfn) (the preferred way)
* get_func_frame(tinfo_t *out, const func_t *pfn)

Removed functions

* get_stkvar
* get_frame
* get_frame_member_by_id
* get_min_spd_ea
* delete_unreferenced_stkvars
* delete_wrong_stkvar_ops

Added functions

* get_func_frame(out: "tinfo_t",pfn: "func_t const *") -> "bool"
* add_frame_member(pfn: "func_t const *", name: "char const *", offset: "uval_t", tif: "tinfo_t", repr: "value_repr_t"=None, etf_flags: "uint"=0) -> "bool"
* is_anonymous_member_name(name: "char const *") -> "bool"
* is_dummy_member_name(name: "char const *") -> "bool"
* is_special_frame_member(tid: "tid_t") -> "bool"
* set_frame_member_type(pfn: "func_t const *",offset: "uval_t", tif: "tinfo_t", repr: "value_repr_t"=None, etf_flags: "uint"=0) -> "bool"

```
* delete_frame_members(pfn: "func_t const *",start_offset: "uval_t", end_offset: "uval_t") -> "bool"
* calc_frame_offset(pfn: "func_t *", off: "sval_t", insn: "insn_t const *"=None, op: "op_t const *"=None) -> "sval_t"

## Modified functions

| 8.4
| 9.0
|
|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| define_stkvar(pfn: "func_t *", name: "const char *", off: "sval_t", flags: "flags64_t", ti: "const opinfo_t *", nbytes: "asize_t") -> bool | define_stkvar(pfn: "func_t *", name: "char const *", off: "sval_t", tif: "tinfo_t", repr: "value_repr_t"=None) -> "bool" |


# ida_bytes

## Removed functions

* free_chunck
* bin_search
* get_8bits

## Added functions

* find_bytes(bs: typing.Union[bytes, bytearray, str],
        range_start: int,
        range_size: typing.Optional[int] = None,
        range_end: typing.Optional[int] = ida_idaapi.BADADDR,
        mask: typing.Optional[typing.Union[bytes, bytearray]] = None,
        flags: typing.Optional[int] = BIN_SEARCH_FORWARD | BIN_SEARCH_NOSHOW,
        radix: typing.Optional[int] = 16,
        strlit_encoding: typing.Optional[typing.Union[int, str]] = PBSENC_DEF1BPU) -> int


* find_string(_str: str,
        range_start: int,
        range_end: typing.Optional[int] = ida_idaapi.BADADDR,
        range_size: typing.Optional[int] = None,
        strlit_encoding: typing.Optional[typing.Union[int, str]] = PBSENC_DEF1BPU,
        flags: typing.Optional[int] = BIN_SEARCH_FORWARD | BIN_SEARCH_NOSHOW) -> int

## Modified functions

| 8.0
| 9.0
|
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| op_enum(ea: "ea_t", n: "int", id: "enum_t", serial: "uchar"=0) -> "bool"
| op_enum(ea: "ea_t", n: "int", id: "tid_t", serial: "uchar"=0) -> "bool"
|
| get_enum_id(ea: "ea_t", n: "int") -> "tid_t"
| get_enum_id(ea: "ea_t", n: "int") -> "enum_t"
|
| parse_binpat_str(out: "compiled_binpat_vec_t", ea: "ea_t", _in: "char const *", radix: "int", strlits_encoding: "int"=0) -> "str" | parse_binpat_str(out: "compiled_binpat_vec_t", ea: "ea_t", _in: "char const *", radix: "int", strlits_encoding: "int"=0) -> "bool" |
| bin_search3(start_ea: "ea_t", end_ea: "ea_t", data: "compiled_binpat_vec_t", flags: "int) -> ea_t
| bin_search3(start_ea: "ea_t", end_ea: "ea_t", data: "compiled_binpat_vec_t const &", flags: "int") -> ea_t                    |
|
```

| bin_search3(start_ea: "ea_t", end_ea: "ea_t", image: "uchar const *", mask: "uchar const *", len: "size_t", flags: "int") -> ea_t |
| get_octet2(ogen: "octet_geenrator_t") -> "uchar_t*"
| get_octet(ogen: "octet_geenrator_t") -> "uchar_t*"
|

idc

Removed functions

* find_text see [ida_bytes](#ida_bytes)
* find_binary see [ida_bytes](#ida_bytes)

ida_dirtree

Removed functions

* dirtree_cursor_root_cursor
* dirtree_t_errstr

ida_diskio

Removed functions

* enumerate_files2
* eclose

ida_fpro

Added functions

* qflcose(fp: "FILE *") -> "int"

ida_funcs

Added methods

func_item_iterator_t

* set_ea(self, _ea: "ea_t") -> "bool"

ida_gdl

Added classes

edge_t

edgevec_t

node_ordering_t

* clear(self)
* resize(self, n: "int") -> void
* size(self) -> "size_t"
* set(self, _node: "int", num: "int") -> "void"
* clr(self, _node: "int") -> "bool"
* node(self, _order: "size_t") -> "int"
* order(self, _node: "int") -> "int"

ida_graph

Removed classes

node_ordering_t

See [ida-gdl](#ida_gdl)
node_ordering_t has been made an alias of ida_gdl.node_ordering_t

edge_t

See [ida_gdl](#ida_gdl)
edge_t has been made an alias of ida_gdl.edge_t

Renamed clases

8.4	9.0
abstract_graph_t	drawable_graph_t
mutable_graph_t	interactive_graph_t

abstract_graph_t has been made an alias of drawable_graph_t
mutbale_graph_t has been made an alias of interactive_graph_t

Renamed functions

8.4	9.0
create_mutable_graph	create_interactive_graph
delete_mutable_graph	delete_interactive_graph
grcode_create_mutable_graph	grcode_create_interactive_graph

create_mutable_graph has been made an alias of create_interactive_graph
delete_mutable_graph has been made an alias of delete_interactive_graph
grcode_create_mutable_graph has been made an alias of grcode_create_interactive_graph

ida_ida

Added classes

idbattr_valmap_t

idbattr_info_t

* is_node_altval(self) -> "bool"
* is_node_supval(self) -> "bool"
* is_node_valobj(self) -> "bool"
* is_node_blob(self) -> "bool"
* is_node_var(self) -> "bool"
* is_struc_field(self) -> "bool"
* is_cstr(self) -> "bool"
* is_qstring(self) -> "bool"
* is_bytearray(self) -> "bool"
* is_buf_var(self) -> "bool"
* is_decimal(self) -> "bool"
* is_hexadecimal(self) -> "bool"
* is_readonly_var(self) -> "bool"
* is_incremented(self) -> "bool"
* is_val_mapped(self) -> "bool"
* is_hash(self) -> "bool"
* use_hlpstruc(self) -> "bool"
* is_bitmap(self) -> "bool"
* is_onoff(self) -> "bool"

* is_scalar_var(self) -> "bool"
* is_bitfield(self) -> "bool"
* is_boolean(self) -> "bool"
* has_individual_node(self) -> "bool"
* str_true(self) -> "char const *"
* str_false(self) -> "char const *"
* ridx(self) -> "size_t"
* hashname(self) -> "char const *"

ida_idaapi

Removed functions

* get_inf_structure
* loader_input_t_from_linput
* loader_input_t_from_capsule
* loader_input_t_from_fp

ida_idd

Added functions

* cpu2ieee(ieee_out: "fpvalue_t *", cpu_fpval: "void const *", size: "int") -> "int"
* ieee2cpu(cpu_fpval: "void *", ieee_out: "fpvalue_t const &", size: "int") -> "int"

ida_idp

Removed methods

_processor_t

* has_realcvt

processor_t

* get_uFlag

Modified methods

_processor_t

8.4	9.0
gen_stkvar_def(ctx: "outctx_t &", mptr: "member_t const *", v: : "sval_t") -> ssize_t	gen_stkvar_def2(ctx: "outctx_t &", mptr: "udm_t", v: "sval_t", tid: "tid_t") -> "ssize_t"

IDP_Hooks

ev_gen_stkvar_def(self, *args) -> "int" ev_obsolete3(self) -> "int"

Added methods

IDP_Hooks

* ev_gen_stkvar_def2(self, outctx: "outctx_t *", stkvar: "udm_t", v: "sval_t", tid: "tid_t") -> "int"

IDB_Hooks

* lt_udm_created(self, udtname: "char const *", udm: "udm_t") -> "void"
* lt_udm_deleted(self, udtname: "char const *", udm_tid: "tid_t", udm: "udm_t") -> "void"

* lt_udm_renamed(self, udtname: "char const *", udm: "udm_t", oldname: "char const *") -> "void"
* lt_udm_changed(self, udtname: "char const *", udm_tid: "tid_t", udmold: "udm_t", udmnew: "udm_t") -> "void"
* lt_udt_expanded(self, udtname: "char const *", udm_tid: "tid_t", delta: "adiff_t") -> "void"
* frame_created(self, func_ea: "ea_t") -> "void"
* frame_udm_created(self, func_ea: "ea_t", udm: "udm_t") -> "void"
* frame_udm_deleted(self, func_ea: "ea_t", udm_tid: "tid_t", udm: "udm_t") -> "void"
* frame_udm_renamed(self, func_ea: "ea_t", udm: "udm_t", oldname: "char const *") -> "void"
* frame_udm_changed(self, func_ea: "ea_t", udm_tid: "tid_t", udmold: "udm_t", udmnew: "udm_t") -> "void"
* frame_expanded(self, func_ea: "ea_t", udm_tid: "tid_t", delta: "adiff_t") -> "void"

Removed functions

All the _processor_t functions have been removed from ida_idp.

ida_ieee

Removed methods

fpvalue_t

* _get_10bytes
* _set_10bytes

ida_kernwin

Removed functions

* place_t_as_enumplace_t
* place_t_as_structplace_t
* open_enums_window
* open_structs_window
* choose_struc
* choose_enum(title, default_id) -> "enum_t"
* choose_enum_by_value(title, default_id, value, nbytes) -> "enum_t"

Modified function

* place_t_as_idaplace_t has been made an alias of place_t.as_idaplace_t
* place_t_as_simpleline_place_t has been made an alias of place_t.as_simpleline_place_t
* place_t_as_tiplace_t has been made an alias of place_t.as_tiplace_t

Removed classes

* enumplace_t
* structplace_t

Removed methods

place_t

* as_enumplace_t
* as_structplace_t

twinpos_t

* place_as_enumplace_t
* place_as_structplace_t

tagged_line_sections_t

* find_in

Added methods

tagged_line_sections_t

* nearest_before(self, range: "tagged_line_section_t", start: "cpidx_t", tag: "color_t"=0) -> "tagged_line_section_t const *"
* nearest_after(self, range: "tagged_line_section_t", start: "cpidx_t", tag: "color_t"=0) -> "tagged_line_section_t const *"

chooser_base_t

* has_widget_lifecycle(self) -> "bool"

Added functions

* is_ida_library(path: "char *", pathsize: "size_t", handle: "void **") -> "bool"

ida_lines

Removed functions

* set_user_defined_prefix

ida_moved

Modified functions

* bookmarks_t_mark has been made an alias of bookmarks_t.mark
* bookmarks_t_get_desc has been made an alias of bookmarks_t.get_desc
* bookmarks_t_find_index has been made an alias of bookmarks_t.find_index
* bookmarks_t_size has been made an alias of bookmarks_t.size
* bookmarks_t_erase has been made an alias of bookmarks_t.erase
* bookmarks_t_get_dirtree_id has been made an alias of bookmarks_t.get_dirtree_id
* bookmarks_t_get has been made an alias of bookmarks_t.get

ida_nalt

Removed functions

* validate_idb_names

ida_netnode

Modified functions

* netnode.exist has been made an alias of netnode.exist

ida_pro

Removed functions

* uchar_array_frompointer
* tid_array_frompointer
* ea_array_frompointer
* sel_array_frompointer
* int_pointer_frompointer
* sel_pointer_frompointer
* ea_pointer_frompointer

See [Added classes](#added-classes-2) below

Added classes

plugin_options_t

* erase(self, name: "char const *") -> "bool"

uchar_pointer

* assign(self, value: "uchar") -> "void"
* value(self) -> "uchar"
* cast(self) -> "uchar *"
* frompointer(t: "uchar *") -> "uchar_pointer *"

ushort_pointer

* assign(self, value: "ushort") -> "void"
* value(self) -> "ushort"
* cast(self) -> "ushort *"
* frompointer(t: "ushort *") -> "ushort_pointer *"

uint_pointer

* assign(self, value: "uint") -> "void"
* value(self) -> "uint"
* cast(self) -> "uint *"
* frompointer(t: "uint *") -> "uint_pointer *"

sint8_pointer

* assign(self, value: "sint8") -> "void"
* value(self) -> "sint8"
* cast(self) -> "sint8 *"
* frompointer(t: "sint8 *") -> "sint8_pointer *"

int8_pointer

* assign(self, value: "int8") -> "void"
* value(self) -> "int8"
* cast(self) -> "int8 *"
* frompointer(t: "int8 *") -> "int8_pointer *"

uint8_pointer

* assign(self, value: "uint8") -> "void"
* value(self) -> "uint8"
* cast(self) -> "uint8 *"
* frompointer(t: "uint8 *") -> "uint8_pointer *"

int16_pointer

* assign(self, value: "int16") -> "void"
* value(self) -> "int16"
* cast(self) -> "int16 *"
* frompointer(t: "int16 *") -> "int16_pointer *"

uint16_pointer

* assign(self, value: "uint16") -> "void"
* value(self) -> "uint16"
* cast(self) -> "uint16 *"
* frompointer(t: "uint16 *") -> "uint16_pointer *"

int32_pointer

* assign(self, value: "int32") -> "void"
* value(self) -> "int32"
* cast(self) -> "int32 *"
* frompointer(t: "int32 *") -> "int32_pointer *"

uint32_pointer

* assign(self, value: "uint32") -> "void"
* value(self) -> "uint32"
* cast(self) -> "uint32 *"
* frompointer(t: "uint32 *") -> "uint32_pointer *"

int64_pointer

* assign(self, value: "int64") -> "void"
* value(self) -> "int64"
* cast(self) -> "int64 *"
* frompointer(t: "int64 *") -> "int64_pointer *"

uint64_pointer

* assign(self, value: "uint64") -> "void"
* value(self) -> "uint64"
* cast(self) -> "uint64 *"
* frompointer(t: "uint64 *") -> "uint64_pointer *"

ssize_pointer

* assign(self, value: "ssize_t") -> "void"
* value(self) -> "ssize_t"
* cast(self) -> "ssize_t *"
* frompointer(t: "ssize_t *") -> "ssize_pointer *"

bool_pointer

* assign(self, value: "bool") -> "void"
* value(self) -> "bool"
* cast(self) -> "bool *"
* frompointer(t: "bool *") -> "bool_pointer *"

short_pointer

* assign(self, value: "short") -> "void"
* value(self) -> "short"
* cast(self) -> "short *"
* frompointer(t: "short *") -> "short_pointer *"

char_pointer

* assign(self, value: "char") -> "void"
* value(self) -> "char"
* cast(self) -> "char *"
* frompointer(t: "char *") -> "char_pointer *"

sel_pointer

* assign(self, value: "sel_t") -> "void"
* value(self) -> "sel_t"
* cast(self) -> "sel_t *"
* frompointer(t: "sel_t *") -> "sel_pointer *"

asize_pointer

* assign(self, value: "asize_t") -> "void"

* value(self) -> "asize_t"
* cast(self) -> "asize_t *"
* frompointer(t: "asize_t *") -> "asize_pointer *"

adiff_pointer

* assign(self, value: "adiff_t") -> "void"
* value(self) -> "adiff_t"
* cast(self) -> "adiff_t *"
* cast(self) -> "adiff_t *"

uval_pointer

* assign(self, value: "uval_t") -> "void"
* value(self) -> "uval_t"
* cast(self) -> "uval_t *"
* frompointer(t: "uval_t *") -> "uval_pointer *"

ea32_pointer

* assign(self, value: "ea32_t") -> "void"
* value(self) -> "ea32_t"
* cast(self) -> "ea32_t *"
* frompointer(t: "ea32_t *") -> "ea32_pointer *"

ea64_pointer

* assign(self, value: "ea64_t") -> "void"
* value(self) -> "ea64_t"
* cast(self) -> "ea64_t *"
* frompointer(t: "ea64_t *") -> "ea64_pointer *"

flags_pointer

* assign(self, value: "flags_t") -> "void"
* value(self) -> "flags_t"
* cast(self) -> "flags_t *"
* frompointer(t: "flags_t *") -> "flags_pointer *"

flags64_pointer

* assign(self, value: "flags64_t") -> "void"
* value(self) -> "flags64_t"
* cast(self) -> "flags64_t *"
* frompointer(t: "flags64_t *") -> "flags64_pointer *"

tid_pointer

* assign(self, value: "tid_t") -> "void"
* value(self) -> "tid_t"
* cast(self) -> "tid_t *"
* frompointer(t: "tid_t *") -> "tid_pointer *"

Added functions

* get_login_name() -> "qstring *"

ida_regfinder

Removed functions

* reg_value_info_t_make_dead_end
* reg_value_info_t_make_aborted
* reg_value_info_t_make_badinsn

* reg_value_info_t_make_unkinsn
* reg_value_info_t_make_unkfunc
* reg_value_info_t_make_unkloop
* reg_value_info_t_make_unkmult
* reg_value_info_t_make_num
* reg_value_info_t_make_initial_sp

Modified functions

8.4	9.0
invalidate_regfinder_cache(ea: "ea_t") -> "void"	invalidate_regfinder_cache(from=BADADDR: "ea_t", to=BADADDR: "ea_t") -> "void"

Added methods

reg_value_info_t

* movt(self, r: "reg_value_info_t", insn: "insn_t const &") -> "void"

ida_registry

Removed functions

* reg_load
* reg_flush

ida_search

Removed functions

* find_binary

ida_ua

Removed Function

* construct_macro(insn: "insn_t *", enable: "bool", build_macro: "PyObject *") -> bool (See [Modified functions](#modified-functions-4))

Modified functions

8.4	9.0
construct_macro2(_this: "macro_constructor_t *", insn: "insn_t *", enable: "bool") -> "bool"	construct_macro(_this: "macro_constructor_t *", insn: "insn_t *", enable: "bool") -> "bool"

Added methods

macro_constructor_t

* construct_macro(self, insn: "insn_t", enable: "bool") -> "bool"

idautils

Modified functions

8.4	9.0
Structs() -> [(idx, sid, name)]	Structs() -> [(ordinal, sid, name)]
StructMembers(sid) -> [(offset, name, size)]	StructMembers(sid) -> [(offset_in_bytes, name, size_in_bytes)]

Examples

This section gives examples of how to port some struct and enum functions using ida_typeinf.

del_struct_members

The following code can be used as an example of how to replace ida_struct.del_struct_members.

```python
def del_struct_members(sid, offset1, offset2):
    tif = ida_typeinf.tinfo_t()
    if tif.get_type_by_tid(sid) and tif.is_udt():
        udm = ida_typeinf.udm_t()
        udm.offset = offset1 * 8
        idx1 = tif.find_udm(udm, ida_typeinf.STRMEM_OFFSET)
        udm = ida_typeinf.udm_t()
        udm.offset = offset2 * 8
        idx2 = tif.find_udm(udm, ida_typeinf.STRMEM_OFFSET)
        return tif.del_udms(idx1, idx2)
```

get_best_fit_member

The following code can be used as an example of how to replace ida_struct.get_best_fit_member.

```python
def get_best_fit_member(sid, offset):
    tif = ida_typeinf.tinfo_t()
    if tif.get_type_by_tid(sid) and tif.is_udt():
        udt = ida_typeinf.udt_type_data_t()
        if tif.get_udt_details(udt):
            idx = udt.get_best_fit_member(offset)
            if not idx == -1:
                return udt[idx]
    return None
```

get_innermost_member

The following code can be used as an example of how to replace ida_struct.get_innermost_member.

```python
def get_innermost_member(sid, offset):
    tif = ida_typeinf.tinfo_t()
    if tif.get_type_by_tid(sid) and tif.is_udt():
        (mtif, idx, _) = tif.get_innermost_udm(offset * 8)
        udt = ida_typeinf.udt_type_data_t()
        if not idx == -1:
            if tif.get_udt_details(udt):
                return mtif, udt[idx]
    return None
```

get_member_fullname

The following code can be used as an example of how to replace ida_struct.get_member_fullname.

```python
def get_member_fullname(udm_tid):
    return ida_typeinf.get_tid_name(udm_tid)
```