The ToolTalk Functional Groupings 3

This chapter describes the ToolTalk functions component of the ToolTalk application programming interface (API). The functions are grouped to perform specific operations; for example, the functions required to initialize the ToolTalk Service. They are grouped under the following headings:

- Initialization Functions
- Message Patterns
- Ptypes
- Sessions
- Files
- Messages
- Objects
- ToolTalk Storage Management
- ToolTalk Error Status
- Exiting
- ToolTalk Error-Handling Macros

See Chapter 4, "The ToolTalk Functions," for a detailed description of these functions.

3

3.1 Initialization Functions

Table 3-1 Initializing and Registering with the ToolTalk Service

Return Type	ToolTalk Function
char *	tt_X_session(const char *xdisplay)
Tt_status	tt_default_session_set(const char *sessid)
char *	tt_open(void)
char *	tt_default_procid(void)
Tt_status	tt_default_procid_set(const char *procid)
Tt_status	tt_ptype_declare(const char *ptid)
Tt_status	tt_ptype_undeclare(const char *ptid)
Tt_status	tt_ptype_exists(const char *ptid)
char *	tt_default_ptype(void)
Tt_status	tt_default_ptype_set(const char *ptid)
int	tt_fd(void)

3.2 Message Patterns

Table 3-2 Creating, Filling In, Registering, and Destroying Message Patterns

Return Type	ToolTalk Function
Tt_pattern	tt_pattern_create(void)
Tt_status	tt_pattern_arg_add(Tt_pattern p, Tt_mode n, const char *vtype, const char *value)
Tt_status	tt_pattern_barg_add(Tt_pattern m, Tt_mode n, const char *vtype, const unsigned char *value, int len)
Tt_status	tt_pattern_iarg_add(Tt_pattern m, Tt_mode n, const char *vtype, int value)
Tt_status	tt_pattern_xarg_add(Tt_pattern m, Tt_mode n, const char *vtype, xdrproc_t xdr_proc, void *value)
Tt_status	tt_pattern_address_add(Tt_pattern p, Tt_address d)

Table 3-2 Creating, Filling In, Registering, and Destroying Message Patterns (Continued)

Return Type	ToolTalk Function
Tt_status	tt_pattern_callback_add(Tt_pattern m, Tt_message_callback f)
Tt_category	tt_pattern_category(Tt_pattern p)
Tt_status	tt_pattern_category_set(Tt_pattern p, Tt_category c)
Tt_status	tt_pattern_class_add(Tt_pattern p, Tt_class c)
Tt_status	tt_pattern_bcontext_add(Tt_pattern p, const char *slotname, const unsigned char *value, int length);
Tt_status	tt_pattern_context_add(Tt_pattern p, const char *slotname, const char *value);
Tt_status	tt_pattern_icontext_add(Tt_pattern p, const char *slotname, int value);
Tt_status	tt_pattern_xcontext_add(Tt_pattern p, const char *slotname, xdrproc_t xdr_proc, void *value)
Tt_status	tt_pattern_destroy(Tt_pattern p)
Tt_status	tt_pattern_disposition_add(Tt_pattern p, Tt_disposition r)
Tt_status	tt_pattern_file_add(Tt_pattern p, const char *file)
Tt_status	tt_pattern_object_add(Tt_pattern p, const char *objid)
Tt_status	tt_pattern_op_add(Tt_pattern p, const char *opname)
Tt_status	tt_pattern_otype_add(Tt_pattern p, const char *otype)
Tt_status	tt_pattern_scope_add(Tt_pattern p, Tt_scope s)
Tt_status	tt_pattern_sender_add(Tt_pattern p, const char *procid)
Tt_status	tt_pattern_sender_ptype_add(Tt_pattern p, const char *ptid)
Tt_status	tt_pattern_session_add(Tt_pattern p, const char *sessid)
Tt_status	tt_pattern_state_add(Tt_pattern p, Tt_state s)
void *	tt_pattern_user(Tt_pattern p, int key)
Tt_status	tt_pattern_user_set(Tt_pattern p, int key, void *v)
Tt_status	tt_pattern_register(Tt_pattern p)
Tt_status	tt_pattern_unregister(Tt_pattern p)
Tt_status	tt_bcontext_join(const char *slotname, const unsigned *char value, int length);
Tt_status	tt_context_join(const char *slotname, const char *value);
Tt_status	tt_icontext_join(const char *slotname, int value);

Table 3-2 Creating, Filling In, Registering, and Destroying Message Patterns (Continued)

Return Type	ToolTalk Function
Tt_status	tt_xcontext_join(const char *slotname, xdrproc_t xdr_proc, void *value);
Tt_status	tt_bcontext_quit(const char *slotname, const unsigned *char value, int length);
Tt_status	tt_xcontext_quit(const char *slotname, xdrproc_t xdr_proc, void *value)
Tt_status	tt_context_quit(const char *slotname, const char *value);
Tt_status	tt_icontext_quit(const char *slotname, int value);

3.3 Ptypes

Table 3-3 Declaring, Undeclaring, and Checking Ptypes

Return Type	ToolTalk Function
Tt_status	tt_ptype_declare(const char *ptid)
Tt_status	tt_ptype_exists(const char *ptid)
Tt_status	tt_ptype_undeclare(const char *ptid)

3.4 Sessions

Table 3-4 Expressing Interest in Sessions

Return Type	ToolTalk Function
char *	tt_default_session(void)
Tt_status	tt_default_session_set(const char *sessid)
char *	tt_initial_session(void)
Tt_status	tt_session_join(const char *sessid)
Tt_status	tt_session_quit(const char *sessid)

Table 3-5 Managing Session Information

Return Type	ToolTalk Function
char *	tt_session_prop(const char *sessid, const char *propname, int i)
Tt_status	tt_session_prop_add(const char *sessid, const char *propname, const char *value)
int	tt_session_prop_count(const char *sessid, const char *propname)
Tt_status	tt_session_prop_set(const char *sessid, const char *propname, const char *value)
Tt_status	tt_session_bprop(const char *sessid, const char *propname, int i, unsigned char **value, int *length)
Tt_status	tt_session_bprop_add(const char *sessid, const char *propname, const unsigned char *value, int length)
Tt_status	tt_session_bprop_set(const char *sessid, const char *propname, const unsigned char *value, int length)
char *	tt_session_propname(const char *sessid, int n)
int	tt_session_propnames_count(const char *sessid)
Tt_status	tt_session_types_load(const char *session, const char *filename)

3.5 Files

Table 3-6 Expressing Interest in Files

Return Type	ToolTalk Function
Tt_status	tt_file_join(const char *filepath)
Tt_status	tt_file_quit(const char *filepath)
char *	tt_default_file(void)
Tt_status	tt_default_file_set(const char *docid)

Table 3-7 Managing Files

Return Type	ToolTalk Function
Tt_status	tt_file_move(const char *oldfilepath, const char *newfilepath)
Tt_status	tt_file_copy(const char *oldfilepath, const char *newfilepath)
Tt_status	tt_file_destroy(const char *filepath)

3.6 Messages

Table 3-8 Creating Messages

Return Type	ToolTalk Function
Tt_message	tt_onotice_create(const char *objid, const char *op)
Tt_message	tt_orequest_create(const char *objid, const char *op)
Tt_message	tt_pnotice_create(Tt_scope scope, const char *op)
Tt_message	tt_prequest_create(Tt_scope scope, const char *op)
Tt_message	tt_message_create(void)
Tt_message	tt_message_create_super(Tt_message m)

Table 3-9 Filling In Messages and Replies

Return Type	ToolTalk Function
Tt_status	tt_message_address_set(Tt_message m, Tt_address p)
Tt_status	tt_message_accept(Tt_message m);
Tt_status	tt_message_arg_add(Tt_message m, Tt_mode n, const char *vtype, const char *value)
Tt_status	tt_message_arg_bval_set(Tt_message m, int n, unsigned char *value, int len)
Tt_status	tt_message_arg_ival_set(Tt_message m, int n, int value)
Tt_status	tt_message_arg_val_set(Tt_message m, int n, const char *value)
Tt_status	tt_message_arg_xval(Tt_message m, int n, xdrproc_t xdr_proc, void *value)

Table 3-9 Filling In Messages and Replies (Continued)

Return Type	ToolTalk Function
Tt_status	tt_message_arg_xval_set(Tt_message m, int n, xdrproc_t xdr_proc, void *value)
Tt_status	tt_message_barg_add(Tt_message m, Tt_mode n, const char *vtype, const unsigned char *value, int len)
Tt_status	tt_message_bcontext_set(Tt_message m, const char *slotname, unsigned char *value, int length);
Tt_status	tt_message_callback_add(Tt_message m, Tt_message_callback f)
Tt_status	tt_message_class_set(Tt_message m, Tt_class c)
int	tt_message_contexts_count(Tt_message m);
Tt_status	tt_message_context_set(Tt_message m, const char *slotname, const char *value);
char *	tt_message_context_slotname(Tt_message m, int n);
Tt_status	tt_message_context_bval(Tt_message m, const char *slotname, unsigned char **value, int *len);
Tt_status	tt_message_context_ival(Tt_message m, const char *slotname, int *value);
char *	tt_message_context_val(Tt_message m, const char *slotname);
Tt_status*	tt_message_context_xval(Tt_message m, const char *slotname, xdrproc_t xdr_proc, void *value);
Tt_status	tt_message_disposition_set(Tt_message m, Tt_disposition r)
Tt_status	tt_message_file_set(Tt_message m, const char *file)
Tt_status	tt_message_handler_ptype_set(Tt_message m, const char *ptid)
Tt_status	tt_message_handler_set(Tt_message m, const char *procid)
Tt_status	tt_message_iarg_add(Tt_message m, Tt_mode n, const char *vtype, int value)
Tt_status	tt_message_icontext_set(Tt_message m, const char *slotname, int value);
Tt_status	tt_message_object_set(Tt_message m, const char *objid)
Tt_status	tt_message_op_set(Tt_message m, const char *opname)
Tt_status	tt_message_otype_set(Tt_message m, const char *otype)
Tt_status	tt_message_scope_set(Tt_message m, Tt_scope s)
Tt_status	tt_message_send_on_exit(Tt_message m);
Tt_status	tt_message_sender_ptype_set(Tt_message m, const char *ptid)

Table 3-9 Filling In Messages and Replies (Continued)

Return Type	ToolTalk Function
Tt_status	tt_message_session_set(Tt_message m, const char *sessid)
Tt_status	tt_message_status_set(Tt_message m, int status)
Tt_status	tt_message_status_string_set(Tt_message m, const char *status_str)
Tt_status	tt_message_user_set(Tt_message m, int key, void *v)
Tt_status	tt_message_xarg_add(Tt_message m, Tt_mode n, const char *vtype, xdrproc_t xdr_proc, void *value)
Tt_status	tt_message_xcontext_set(Tt_message m, const char *slotname, xdrproc_t xdr_proc, void *value)
Tt_status	tt_otype_opnum_callback_add(const char *otid,int opnum, Tt_message_callback f)
Tt_status	tt_ptype_opnum_callback_add(const char *ptid, int opnum, Tt_message_callback f)

Table 3-10 Examining Messages

Return Type	ToolTalk Function
Tt_address	tt_message_address(Tt_message m)
Tt_status	tt_message_arg_bval(Tt_message m, int n, unsigned char **value, int *len)
Tt_status	tt_message_arg_ival(Tt_message m, int n, int *value)
Tt_mode	tt_message_arg_mode(Tt_message m, int n)
char *	tt_message_arg_type(Tt_message m, int n)
char *	tt_message_arg_val(Tt_message m, int n)
Tt_status	tt_message_arg_xval(Tt_message m, int n, xdrproc_t xdr_proc, void *value)
int	tt_message_args_count(Tt_message m)
Tt_class	tt_message_class(Tt_message m)
Tt_disposition	tt_message_disposition(Tt_message m)
char *	tt_message_file(Tt_message m)
gid_t	tt_message_gid(Tt_message m)
char *	tt_message_handler(Tt_message m)

Table 3-10 Examining Messages (Continued)

Return Type	ToolTalk Function
char *	tt_message_handler_ptype(Tt_message m)
char *	tt_message_id(Tt_message m)
char *	tt_message_object(Tt_message m)
char *	tt_message_op(Tt_message m)
int	tt_message_opnum(Tt_message m)
char *	tt_message_otype(Tt_message m)
Tt_pattern	tt_message_pattern(Tt_message m)
Tt_scope	tt_message_scope(Tt_message m)
char *	tt_message_sender(Tt_message m)
char *	tt_message_sender_ptype(Tt_message m)
char *	tt_message_session(Tt_message m)
Tt_state	tt_message_state(Tt_message m)
int	tt_message_status(Tt_message m)
char *	tt_message_status_string(Tt_message m)
uid_t	tt_message_uid(Tt_message m)
void *	tt_message_user(Tt_message m, int key)
Tt_status	tt_message_send(Tt_message m)
Tt_status	tt_message_destroy(Tt_message m)

Table 3-11 Sending and Destroying Messages

Return Type	ToolTalk Function
Tt_status	tt_message_send(Tt_message m)
Tt_status	tt_message_destroy(Tt_message m)

Table 3-12 Receiving, Replying to, Rejecting, and Destroying Messages

Return Type	ToolTalk Function
Tt_message	tt_message_receive(void)
Tt_status	tt_message_reply(Tt_message m)
Tt_status	tt_message_reject(Tt_message m)
Tt_status	tt_message_fail(Tt_message m)
int	tt_message_status(Tt_message m)
Tt_status	tt_message_status_set(Tt_message m, int status)
char *	tt_message_status_string(Tt_message m)
Tt_status	tt_message_status_string_set(Tt_message m, const char *status_str)
Tt_status	tt_message_destroy(Tt_message m)

3.7 Objects

Table 3-13 Creating, Moving, and Destroying Objects

Return Type	ToolTalk Function
char *	tt_spec_create(const char *filepath)
Tt_status	tt_spec_prop_add(const char *objid, const char *propname, const char *value)
Tt_status	tt_spec_prop_set(const char *objid, const char *propname, const char *value)
Tt_status	tt_spec_bprop_add(const char *objid, const char *propname, const unsigned char *value, int length)
Tt_status	tt_spec_bprop_set(const char *objid, const char *propname, const unsigned char *value, int length)
Tt_status	tt_spec_type_set(const char *objid, const char *otid)
Tt_status	tt_spec_write(const char *objid)
char *	tt_spec_move(const char *objid, const char *newfilepath)
Tt_status	tt_spec_destroy(const char *objid)

Table 3-14 Using ToolTalk Storage

Return Type	ToolTalk Function
char *	tt_spec_prop(const char *objid, const char *propname, int i)
int	tt_spec_prop_count(const char *objid, const char *propname)
Tt_status	tt_spec_prop_set(const char *objid, const char *propname, const char *value)
Tt_status	tt_spec_bprop(const char *objid, const char *propname, int i, unsigned char **value, int *length)
char *	tt_spec_propname(const char *objid, int n)
int	tt_spec_propnames_count(const char *objid)
char *	tt_spec_type(const char *objid)
char *	tt_spec_file(const char *objid)
Tt_status	tt_spec_write(const char *objid)
Tt_status	tt_file_objects_query(const char *filepath, Tt_filter_function filter, void *context, void *accumulator)
int	tt_objid_equal(const char *objid1, const char *objid2)
char *	tt_objid_objkey(const char *objid)

Table 3-15 Examining Object Type Information

Return Type	ToolTalk Function
char *	tt_otype_base(const char *otype)
char *	tt_otype_derived(const char *otype, int i)
int	tt_otype_deriveds_count(const char *otype)
Tt_mode	tt_otype_hsig_arg_mode(const char *otype, int sig, int arg)
char *	tt_otype_hsig_arg_type(const char *otype, int sig, int arg)
int	tt_otype_hsig_args_count(const char *otype, int sig)
int	tt_otype_hsig_count(const char *otype)
char *	tt_otype_hsig_op(const char *otype, int sig)
int	tt_otype_is_derived(const char *derivedotype, const char *baseotype)

Table 3-15 Examining Object Type Information (Continued)

Return Type	ToolTalk Function
Tt_mode	tt_otype_osig_arg_mode(const char *otype, int sig, int arg)
char *	tt_otype_osig_arg_type(const char *otype, int sig, int arg)
int	tt_otype_osig_args_count(const char *otype, int sig)
int	tt_otype_osig_count(const char *otype)
char *	tt_otype_osig_op(const char *otype, int sig)

3.8 ToolTalk Storage Management

Table 3-16 Managing ToolTalk Storage

Return Type	ToolTalk Function
int	tt_mark(void)
void	tt_release(int mark)
void	tt_free(caddr_t p)
caddr_t	tt_malloc(size_t s)

3.9 ToolTalk Error Status

Table 3-17 Retrieving ToolTalk Error Information

Return Type	ToolTalk Function
Tt_status	tt_int_error(int return_val)
Tt_status	tt_pointer_error(void *pointer)
char *	tt_status_message(Tt_status ttrc)

Table 3-18 Encoding Error Values

Return Type	ToolTalk Function
int	tt_error_int(Tt_status ttrc)
void *	tt_error_pointer(Tt_status ttrc)

3.10 Exiting

Table 3-19 Leaving the ToolTalk Session

Return Type	ToolTalk Function
Tt_status	tt_close(void)

3.11 ToolTalk Error-Handling Macros

Table 3-20 ToolTalk Error-Handling Macros

Return Type	ToolTalk Macro
int	tt_is_err(Tt_status s)
Tt_status	tt_ptr_error(pointer)