Maintaining Application Information

5

Applications that want to receive ToolTalk messages provide information to the ToolTalk service that describes what kind of messages they want to receive. This information, known as message patterns, is provided dynamically either by applications as they run, or through ptype and otype files.

5.1 Installing Application Types

Installing application types is an occasional task; you only need to install type information when an application error condition exists. Ptype and otype files are run through the ToolTalk type compiler at installation time. `tt_type_comp` merges the information into the ToolTalk Types Database. The application then tells the ToolTalk service to read the type information in the database.

To install an application's ptype and otype files, follow these steps:

1. Run `tt_type_comp` on your type file.

```
% tt_type_comp <your-file>
```

`tt_type_comp` runs *your-file* through `cpp`, compiles the type definitions, and merges the information into a ToolTalk Types table. Table 5-1 describes location of the XDR-base format tables; Table 5-2 describes the location of the Classing Engine-base format table.

Table 5-1 XDR-base Format ToolTalk Types Tables

Database	Uses XDR Table
user	~/.tt/types.xdr
system	/etc/tt/types.xdr
network	$TTHOME/etc/tt/types.xdr

Table 5-2 Classing Engine-base Format ToolTalk Types Tables

Database	Uses Classing Engine Table
user	~/.cetables/cetables
system	/etc/cetables/cetables
network	$OPENWINHOME/lib/cetables/cetables

By default, `tt_type_comp` uses the *user* database. To specify another database, use the `-d` option; for example:

```
% tt_type_comp -d user|system|network <your-file>
```

Note – When you run `tt_type_comp` on your ptype or otype files, it first runs `cpp` on the file and then checks the syntax before it places the data into the ToolTalk Types Database format. If syntax errors are found, a message is displayed that indicates the line number of the cpp file. To find the line, enter

`cpp -P` *source-file temp-file*

and view the *temp-file* to find the error on the line reported by `tt_type_comp`.

2. **Force `ttsession` to reread the ToolTalk Types Database.**
   To force `ttsession` to reread the ToolTalk Types Database, see the "Updating the ToolTalk Service" section.

5.2 Examining ToolTalk Type Information

You can examine all type information in a specified ToolTalk Types Database, only the ptype information, or only the otype information. To specify the database you want to examine, use the `-d` option and supply the name of the *user*, *system*, or *network* to indicate the desired database. If the `-d` option is not used, `tt_type_comp` will use the *user* database by default.

- **To examine all the ToolTalk type information in a ToolTalk Types Database, enter the following line**

```
% tt_type_comp -d user|system|network -p
```

The type information will be printed out in source format.

- **To list all ptypes in a ToolTalk Types Database, enter the following line:**

```
% tt_type_comp -d user|system|network -P
```

The names of the ptypes will be printed out in source format.

- **To list all otypes in a ToolTalk Types Database, enter the following line:**

```
% tt_type_comp -d user|system|network -O
```

The names of the otypes will be printed out in source format.

5.3 Removing ToolTalk Type Information

You can remove both ptype and otype information from the ToolTalk Types Databases.

- **Use `tt_type_comp` to remove type information. Enter the following line:**

```
% tt_type_comp -d user|system|network -r type
```

For example, to remove a ptype called *EditDemo* from the ToolTalk Types network database of a sample application, enter the line:

```
% tt_type_comp -d network -r EditDemo
```

After you remove type information, force any running `ttsessions` to reread the ToolTalk Types Database again to bring the ToolTalk service up-to-date. See "Updating the ToolTalk Service" for more information.

5.4 Updating the ToolTalk Service

When you make changes to the ToolTalk Types Database, you must force any ToolTalk sessions that are already running to reread it or the changes will not be in effect. Use `ttsession` to force the ToolTalk service to read the type information.

1. **Enter the `ps` command to find the process identifier (*pid*) of the `ttsession` process**

```
% ps -ef | grep ttsession
```

2. **Enter the `kill` command to send a SIGUSR2 signal to `ttsession`.**

```
% kill -USR2 <ttsession pid>
```

5.5 Process Type Errors

One or both of the following conditions exists if applications report the error:

```
Application is not an installed ptype.
```

1. The ToolTalk service has not been instructed by the application to reread the recently updated type information in the ToolTalk Types Database. See "Updating the ToolTalk Service" for instructions on how to force the ToolTalk service to reread type information from the ToolTalk Types Database.

2. The application's ptypes and otypes have not been compiled and merged into the ToolTalk Types Database. See "Installing Application Types" for instructions on how to compile and merge type information.

5.6 Using ttsnoop to Debug Messages and Patterns

ttsnoop is a tool provided to create and send custom-constructed ToolTalk messages. You can also use *ttsnoop* as a tool to selectively monitor any or all ToolTalk messages. The *ttsnoop* program resides in the directory `$OPENWINHOME/share/src/tooltalk/demo/ttsnoop`. To compile the program, enter the following commands on the command line:

```
cd $OPENWINHOME/share/src/tooltalk/demo/ttsnoop
make
```

Note – To run *ttsnoop* as a debugging tool, you must be running the OpenWindows software. Set the `$OPENWINHOME` environment variable to where the OpenWindows software is installed.

To start the program, enter the following command on the command line:

```
ttsnoop [ -t ]
```

The -t option displays the ToolTalk API calls that are being used to construct a particular pattern or message. Figure 5-1 shows the window that is displayed when *ttsnoop* starts.

5

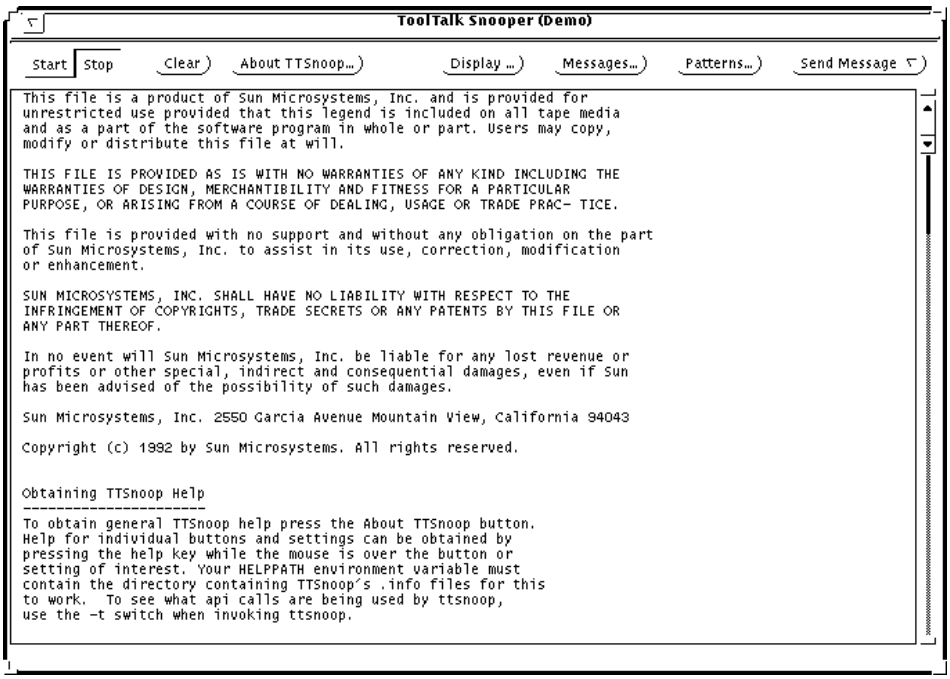

Figure 5-1 ttsnoop Display Window

- **Start**
  Press this button to activate message reception. *ttsnoop* will display any incoming messages which match the patterns you register.

- **Stop**
  Press this button to stop receiving messages.

- **Clear**
  Press this button to clear the window.

- **About TTSnoop**
  Press this button to obtain general help for *ttsnoop*.

Note – To obtain help for individual buttons and settings, place the mouse over the button or setting and press the F1 key or Help key on your keyboard. Your `HELPPATH` environment variable must contain the directory that contains the TTSnoop `.info` files.

- **Display**
  Press this button to popup a panel of checkboxes to highlight specific ToolTalk message components on the *ttsnoop* display subwindow.
- **Messages**
  Press this button to popup a panel which allows you to create, store, and send ToolTalk messages.
- **Patterns**
  Press this button to popup a panel which allows you to compose and register ToolTalk patterns.
- **Send Messages**
  Press this button to send messages that were stored using the Messages popup.

5

Composing and Sending Messages

When you press the Messages button on the initial display window, the popup panel shown in Figure 5-2 is displayed.

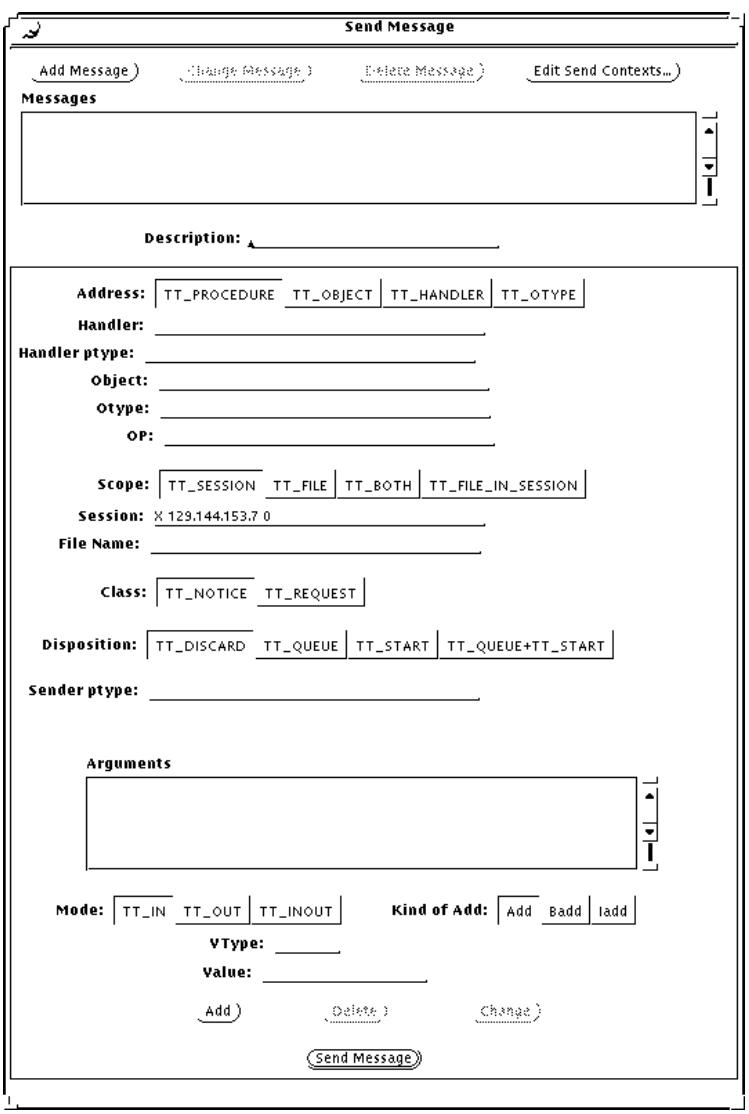

Figure 5-2 Popup Messages Panel

- **Add Message**
  Press this button to store the current message settings. Once the messages are stored, you can recall and send these messages using the Send Message button on the initial display window.

- **Edit Contexts**
  Press this button to add, change, and delete send message contexts. The popup window displayed, shown in Figure 5-3, allows you to edit contexts to be sent with your messages.

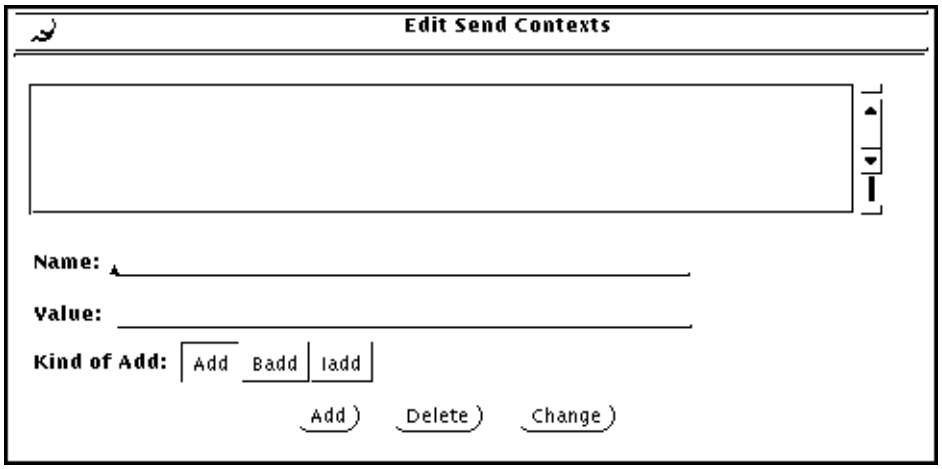

Figure 5-3 Editing Send Contexts

- **Send Message**
  Press this button to send the newly created message.

Note – This button performs the same function as the Send Message button on the main menu.

5

Composing and Registering Patterns

When you press the Patterns button on the initial display window, the popup panel shown in Figure 5-4 is displayed.

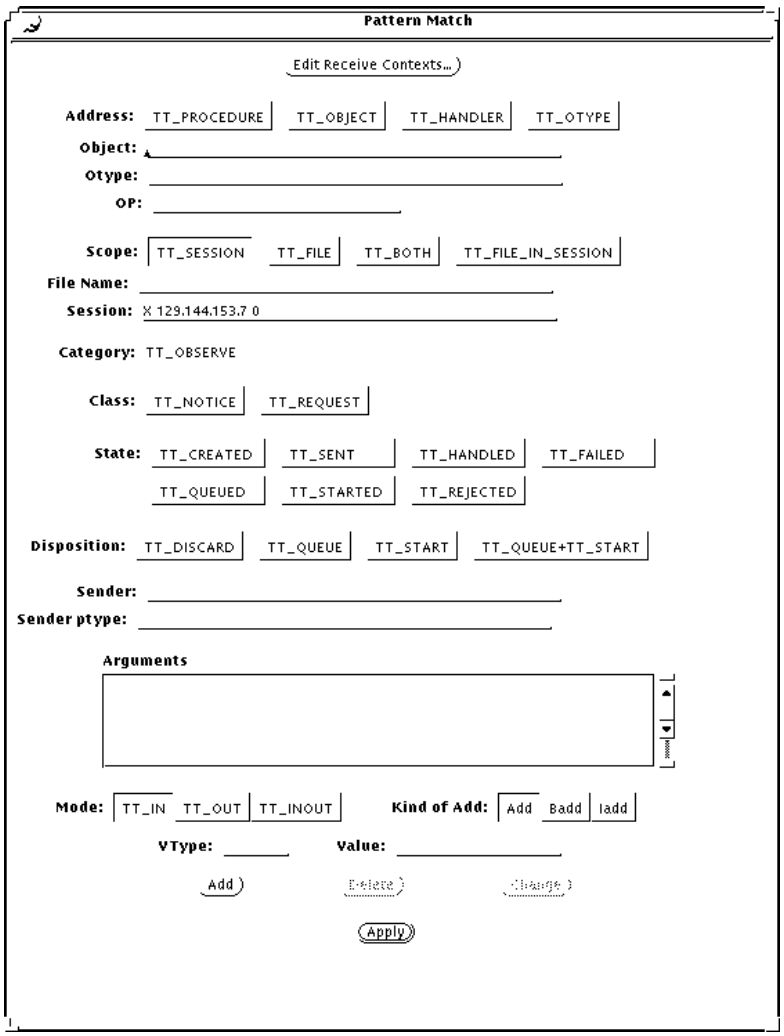

Figure 5-4 Popup Patterns Panel

Press the Apply button to register your pattern. Once a pattern is registered, you can use *ttsnoop* as a debugging tool to observe what messages are being sent by other applications.

- **Edit Receive Contexts**
  Press this button to add, change, and delete receive message contexts in patterns. The popup window displayed, shown in Figure 5-5, allows you to edit contexts to be registered with your patterns.

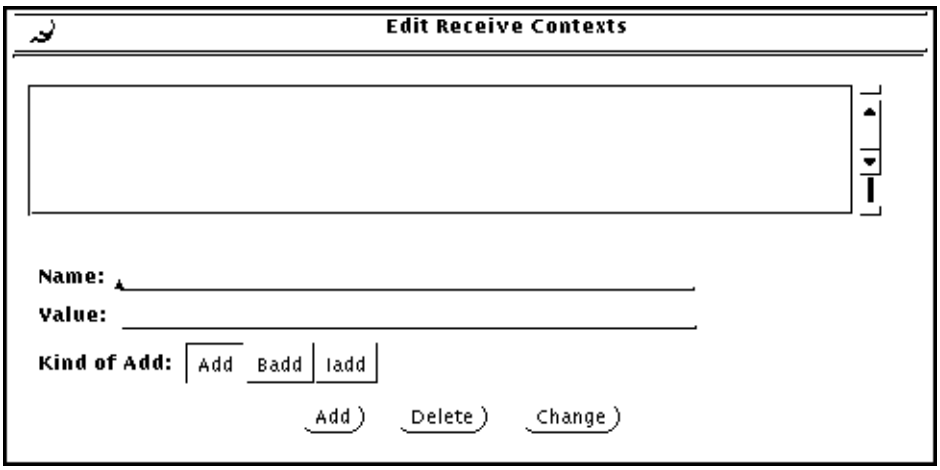

Figure 5-5 Editing Message Contexts in Patterns

Displaying Message Components

When you press the Display button on the initial display window, the popup a panel of checkboxes shown in Figure 5-6 is displayed.

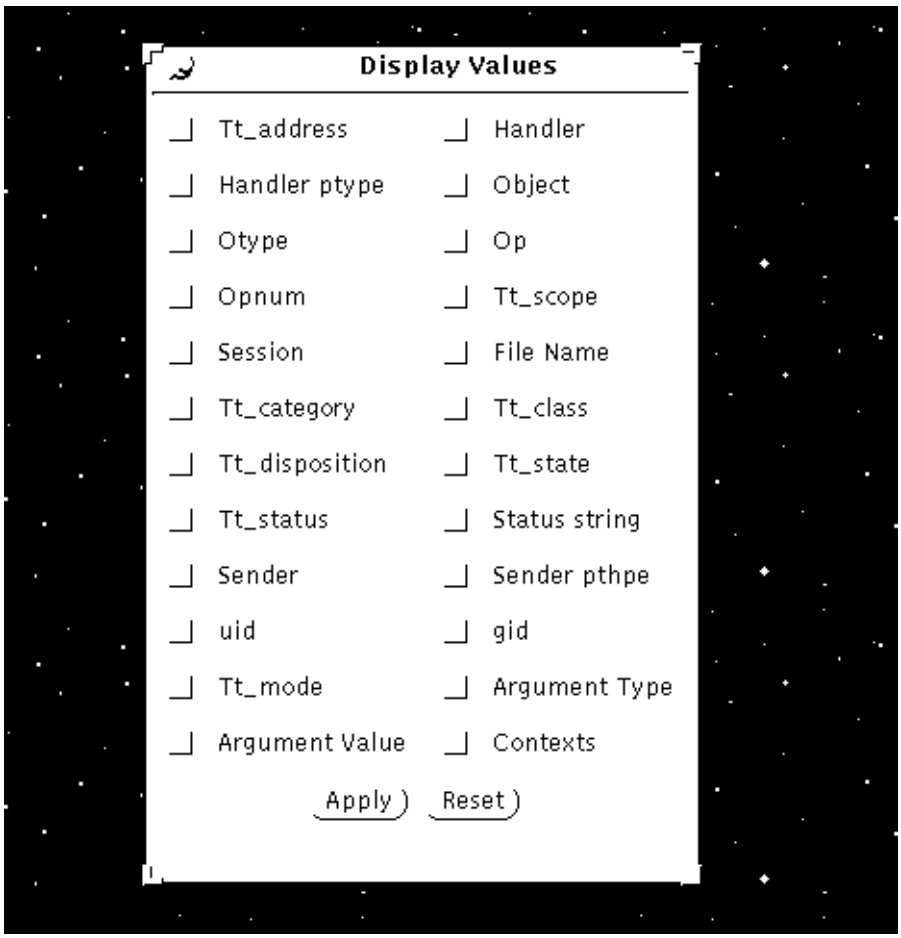

Figure 5-6 Checkboxes to Display Message Component Values

When you select a checkbox, the specified ToolTalk message component is indicated on a displayed message by an arrow (--->) to the left of the displayed message component. Figure 5-7 shows the displayed message components.

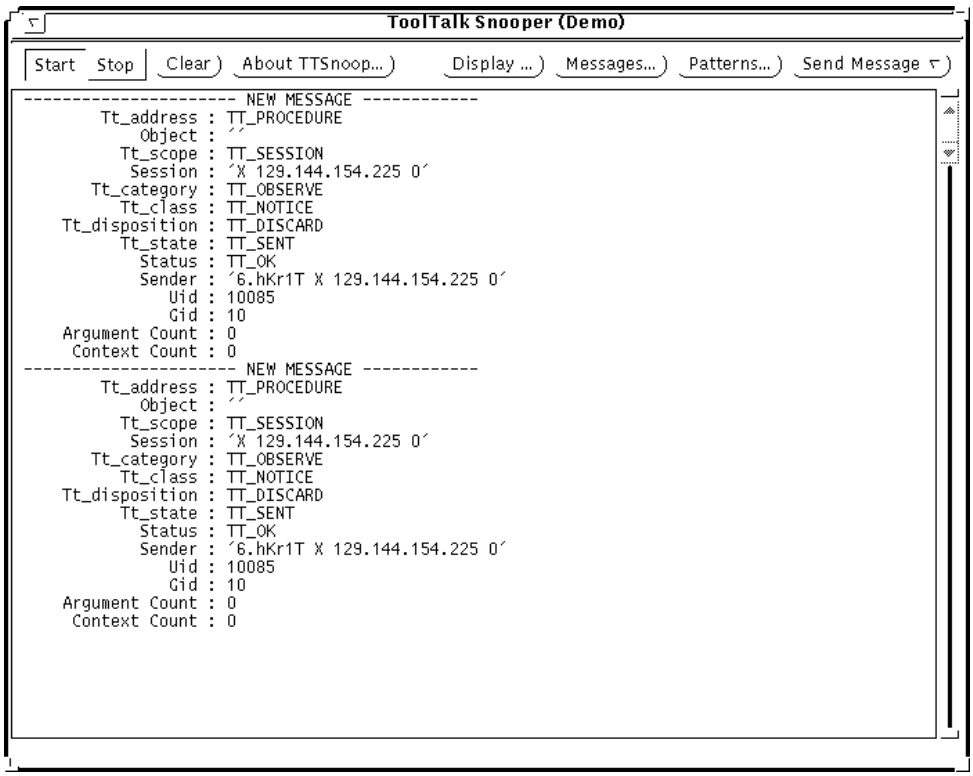

Figure 5-7 Display of Message Components

Sending Pre-Created Messages

When you press the Send Message button on the initial display window, you can send one the messages you created and stored using the Messages popup.

Receiving Messages

When you press the Start button on the initial display window, *ttsnoop* will display any incoming messages which match the patterns you registered. Figure 5-8 is an example of a displayed incoming message.

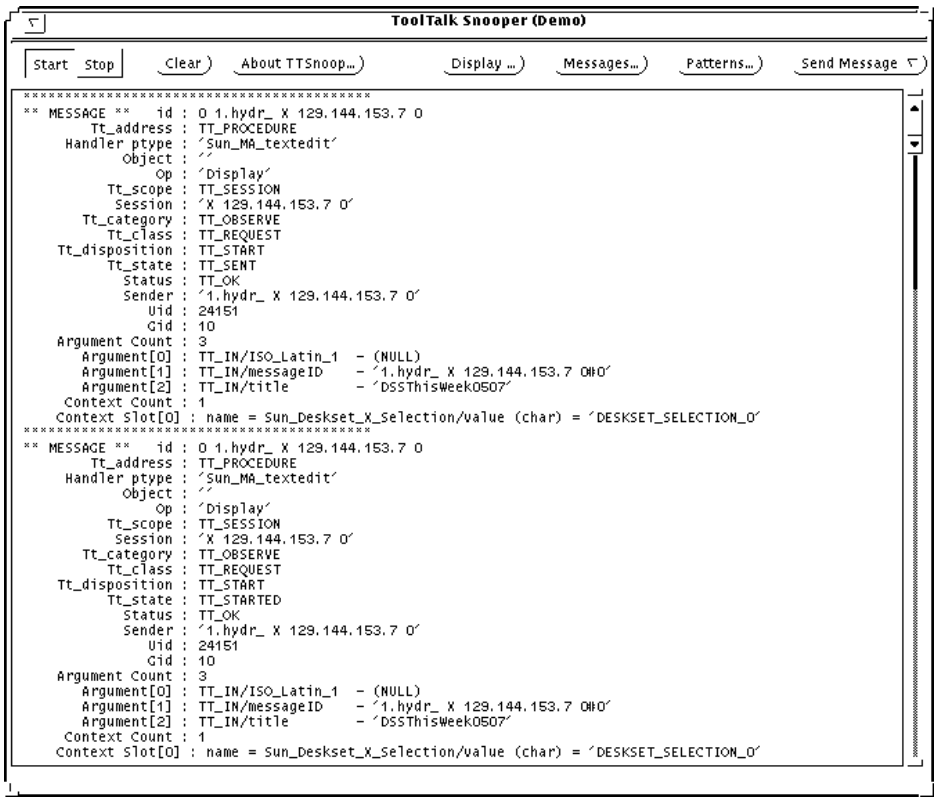

Figure 5-8 Incoming Message Displayed

Stop Receiving Messages

When you press the Stop button on the initial display window, *ttsnoop* will stop receiving messages.