Argus

Getting Started
Draft: 12 February 1990 at 20:31

Wendy E. Mackay

Running Argus for the First Time

In order to run an Argus program, you must first include a reference to the main Argus directory in your Unix path. When you run an Argus program for the first time, it will be a bit slower than normal, because it will create several directories and files for you. These will contain your personal rulesets and (optionally) data about how you use Argus.

The purpose of this document is to provide the minimum information necessary to set up Argus for the first time and test it to make sure that everything works. When you have successfully set up Argus and run a rule, you should refer to the documentation or on-line help for additional information and suggestions for use. Note that the examples given are for Project Athena. Other sites will refer to different directories and may have different set up procedures.

Note: The following examples would work for someone with the username of "smith". Substitute your own username whenever you see "smith".

1. **Add a reference to Argus to your Unix path.** Your Unix path is usually specified in your *.path* file. Edit the appropriate file so that it includes a reference to the Argus executable files, which is /afs/athena/astaff/project/argus/1.0/${hosttype}bin in release 1.0 at Project Athena. For example:

   ```
   set path=(. $athena_path
   /afs/athena/astaff/project/argus/1.0/${hosttype}bin /)
   ```

2. You must now update your path explicitly. The next time you login, this file should be read automatically.

   ```
   source ~/.path
   ```

3. You are now ready to create your first rule. Create a rule that will be stored in a ruleset called TEST.

   ```
   ruleeditor test &
   ```

4. The ruleeditor looks for a ruleset directory, but doesn't find it. So it creates a ruleset directory for you. You should see the following messages:

   ```
   AlRSreg_init(): ruleset directory /mit/smith/argus/rulesets
     does not exist, creating it.
   AlData_init(): ruleset directory /mit/smith/argus/data does
     not exist, creating it.
   ```

5. The ruleeditor will now look in the ruleset directory for a ruleset called "test". Since you haven't created it yet, it will ask you if you would like to create it now. Press the **OK** button.

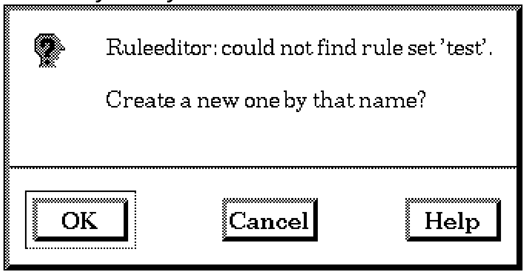

Figure 1: Prompt when creating a new ruleset, called "Test".

6. A ruleeditor will appear, with two empty fields (TO: and SUBJECT:). The title should say: EDITING: rule 1 of rule set 'test'.

To test the ruleeditor, create a rule that looks for messages from you which contain a specified subject. Use a Unix command ("cat") to display the mail message on the screen.

Place the mouse cursor in the empty field to the right of TO: field and click the left mouse button. Then type your username.

Do the same for the empty field to the left of the SUBJECT: field and type a test message, such as "Testing Argus".

Move to the **THEN** section and click the left mouse button on COPY TO. This will produce a menu of other possible actions. Move the mouse down to SHELL WITH MESSAGE AS STDIN and release to select this new action.

Click the left mouse button on the empty field to the right and type "cat". You have now told Argus that, if this rule fires on a particular message, the message will be treated as standard input to the Unix "cat" command and displayed on your screen.

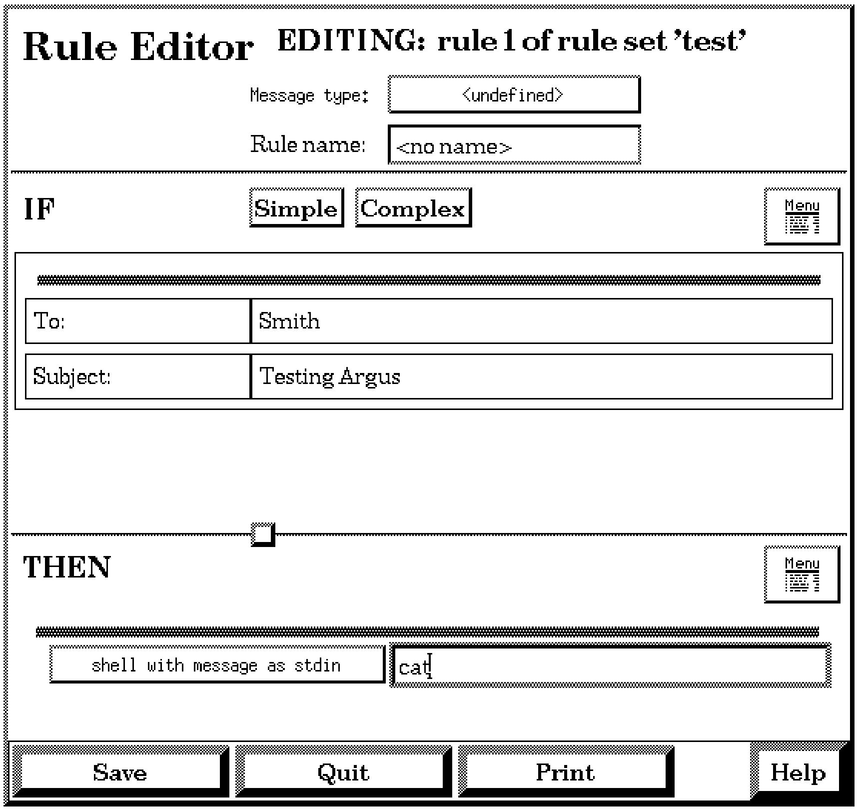

Figure 2: Ruleeditor: Test rule

7. Press the ruleeditor's **Save** button to save the rule.
8. Press the **Quit** button to quit.
9. Test your new rule by creating a message that meets the specified criteria and send it to yourself.
10. Incorporate ("inc") your new mail:

    ```
    inc
    ```

11. Run rules to test that it worked. If you use "mh", the command will look like this: ("test" is the name of your newly created ruleset, "inbox" is the name of your mail folder).

```
runrules test inbox
```

If you use rmail, you must specify "rmail" format. (Mh is the default value.)

```
runrules -MSrmail test inbox
```

12. Your newly-created message should appear in the same window. If your inbox is very large, this may take a while. If your message doesn't appear, make sure your message actually arrived, and that it fits the criteria you specified in the rule. Check that that the subject of your test message matches the subject that you placed in your test rule.

13. Now, use the ruleset browser to see and edit your rule.

```
rulebrowser test &
```

14. The ruleset browser will appear, with "Viewing Rule Set: test" in the title. Only one rule should be visible. You should see the first line (IF To: contains smith [Case Insensitive with Commas as Ors, Message] of the rule.

15. To edit the rule, left click on the **Menu** icon that appears to the left and choose EDIT THIS RULE (also with the left mouse button).

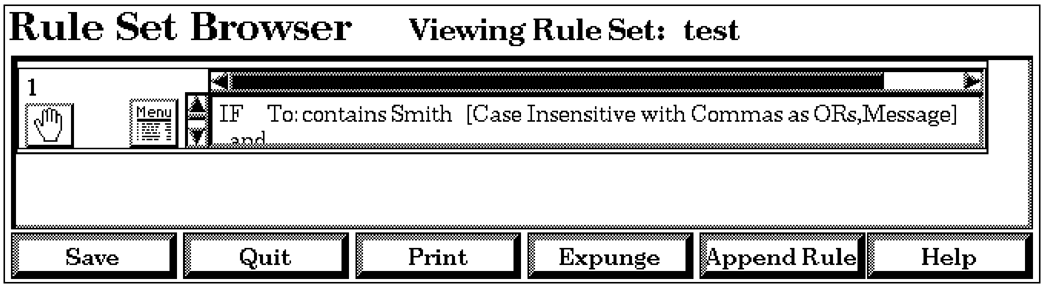

Figure 3: Ruleset Browser: Display of test rule.

16. An icon that says "editing" will appear next to the menu icon. A ruleeditor will appear, with the rule you just created. Edit it so that it pops up a zephyr window on your screen when it finds your message. Click left on the field that previously said "cat", delete the letters, and retype "zwrite" followed by your username. (Note: you must have a zephyr client running for this rule to work. If you don't have zephyr running, try replacing cat with: echo "Testing Argus".)

shell with message as stdin	zwrite smith

17. Press the ruleeditor's **Update** button to save the rule.

18. Press the ruleset browser's **Save** button to save the ruleset.

19. Press the ruleset browser's **Quit** button to leave both the ruleeditor and the ruleset browser.

20. Use the same "runrules" command you used before to check the rule. A zephyr message should appear on the screen, containing the message you created.

21. If you would like to collect data about how you use Argus, run the ***argusi*** program.

```
argusi
```

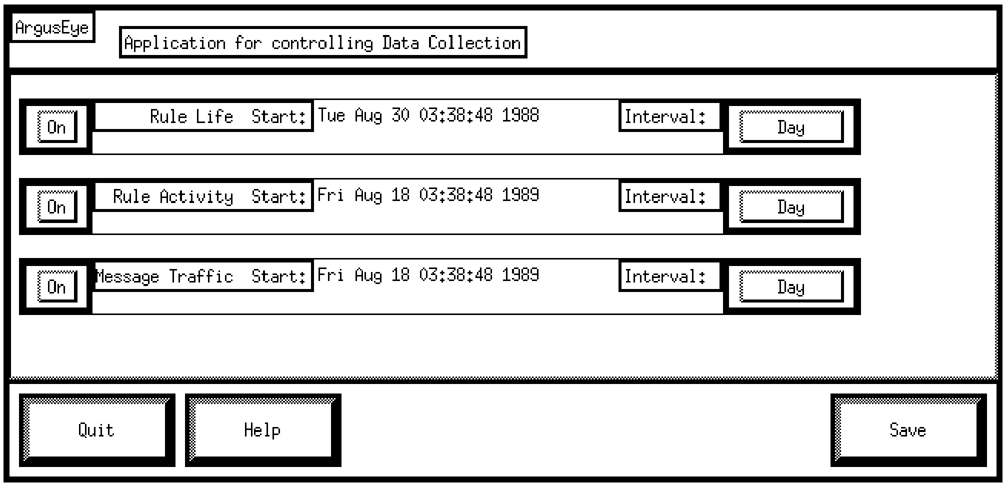

Figure 4: Argusi: Controls whether data is collected about use of rules.

22. You will see three types of data collection: rule life, rule activity, and message traffic. You can turn any or all of them on, and they will begin collecting data from the time specified after "start:", at the intervals you specify. The default values are all "off". If you turn on data collection, the files will be stored in your ~/argus/data directory.

User Interface "Gotcha's"

We have a number of known user interface "bugs"; features that may not work as you expect them to.

These include:

1. If you click on the **menu** icon in the ruleeditor, you'll see that the option "add a text field" is gray'd out. You must click the mouse button in one of the fields in the **IF** section to make this option available.

2. There is a gray bar in the box just above the TO: field, which is called the "control bar". You should be able to click on it directly, but in this release, you must use the **TAB** key to move through the various fields until the control bar is highlighted. The purpose of this bar is to allow you to place a new field at the top of the rule. Otherwise, when you choose "add a text field", it will insert it below the currently selected field.

3. You can group one or more fields in the **IF** section of the rule, by surrounding them with a box and specifying a logical operator. If you choose "group fields with OR" from the **Menu**, the mouse icon will turn into a symbol for an upper left hand corner. Place this at the upper left hand corner of the set of fields you'd like to group and click the left mouse button. The cursor will turn into a symbol for the lower right hand corner. DRAG the mouse to the lower right hand corner of the set of fields and click the left mouse button. The fields will rearrange themselves, with the word "OR" in the upper left hand corner of a new box, with a new control bar and the specified fields below.

4. Selecting the **Menu** icon in the **THEN** section is a bit tricky. First, you must click on one of the **THEN** fields to activate the menu. Then, if you choose "add an action", a second menu of possible actions will come up very quickly. KEEP HOLDING the mouse button, and don't release it until it is over the action you want to choose. If you choose the "add an action" option by accident, move the mouse outside of the menu and release.

5. If the ruleeditor window is near the bottom of your screen, and you choose to edit an existing action by clicking on it, part of the menu of options will be obscured. Try moving the ruleeditor to the top of the screen before making the choice.

6. If you add new fields and the window does not have enough room to display them, scroll bars will appear. You can scroll through to display the fields you are interested in, or resize the entire window. If you would like the adjust the relative amount of space between the **IF** and **Then** sections, place your mouse cursor in the square on the line between the two sections and move it to the desired location.

7. When you call the ruleeditor directly, you'll see **Save** and **Quit**. When you call the ruleeditor from the ruleset browser, these are replaced with **Update Rule** and **Finished**. They are functionally equivalent.

8. In addition to the **HELP** button at the bottom of the ruleeditor, you can also get help from the various menus. Some provide examples as well.

Rule Examples to help you get started

You can identify all messages that are addressed to you personally (as opposed to a distribution list) and treat them separately. Some people like to move these messages into a special "high priority" folder.

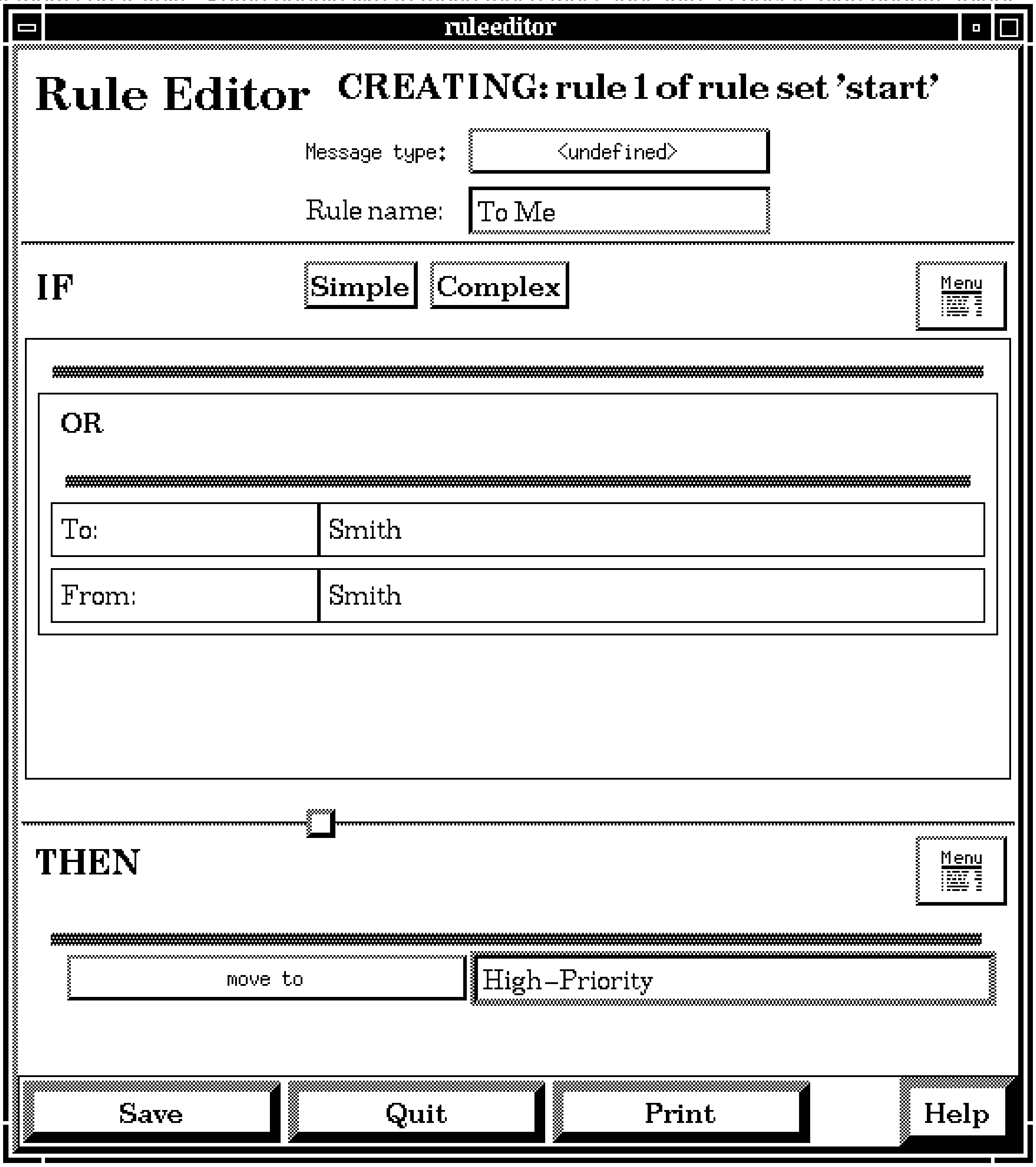

Figure 5: Identify personally-addressed rules

Other people prefer to leave personally-addressed messages in their inboxes and move the other, lower prio ther kno

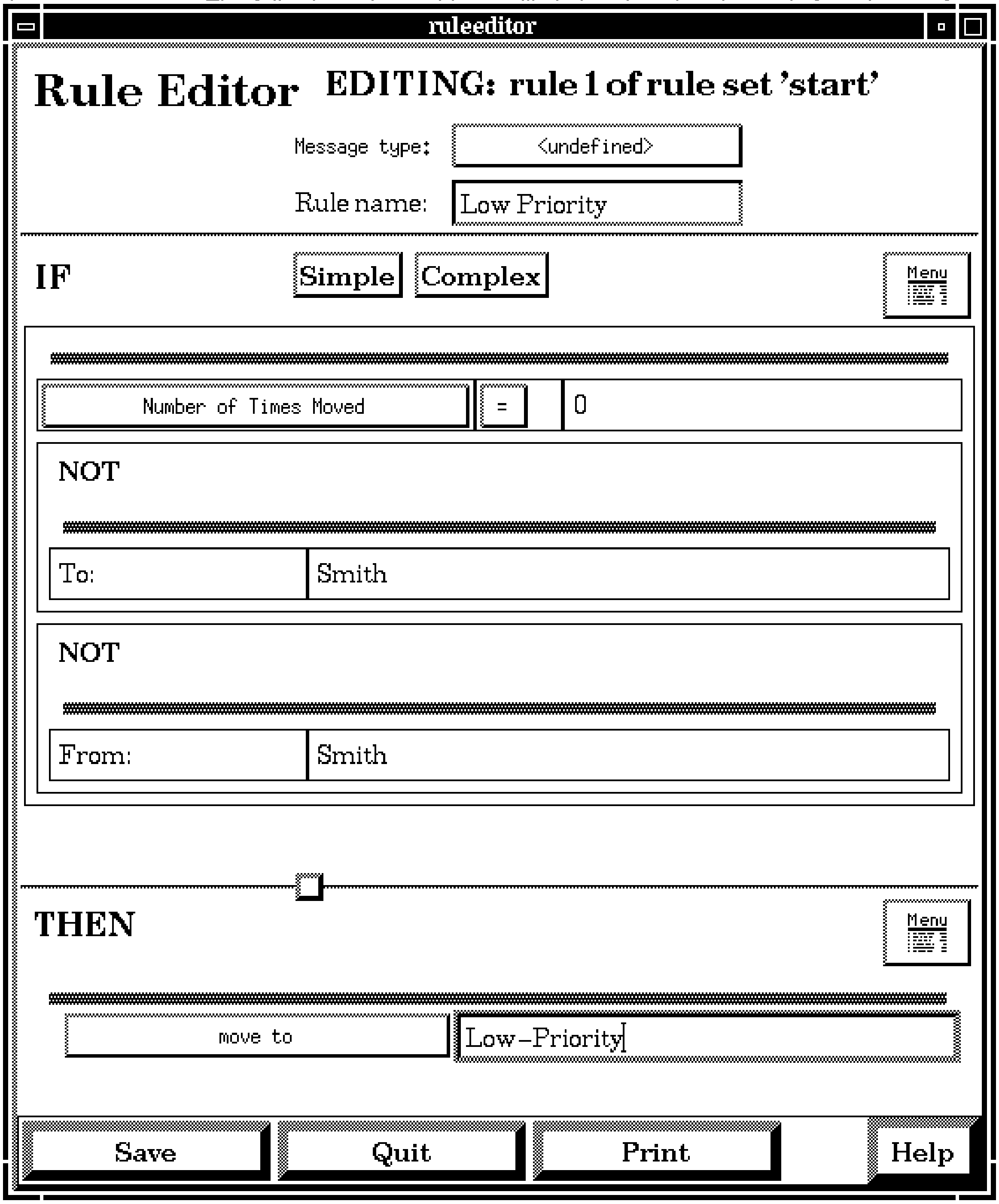

Figure 6: Move non-personal messages to a low priority folder

You can identify and save all messages from a particular distribution group in a specified folder. Note that these messages are usually addressed **to** a distribution group. However, in some mail systems, they may [illegible] your rule [illegible]

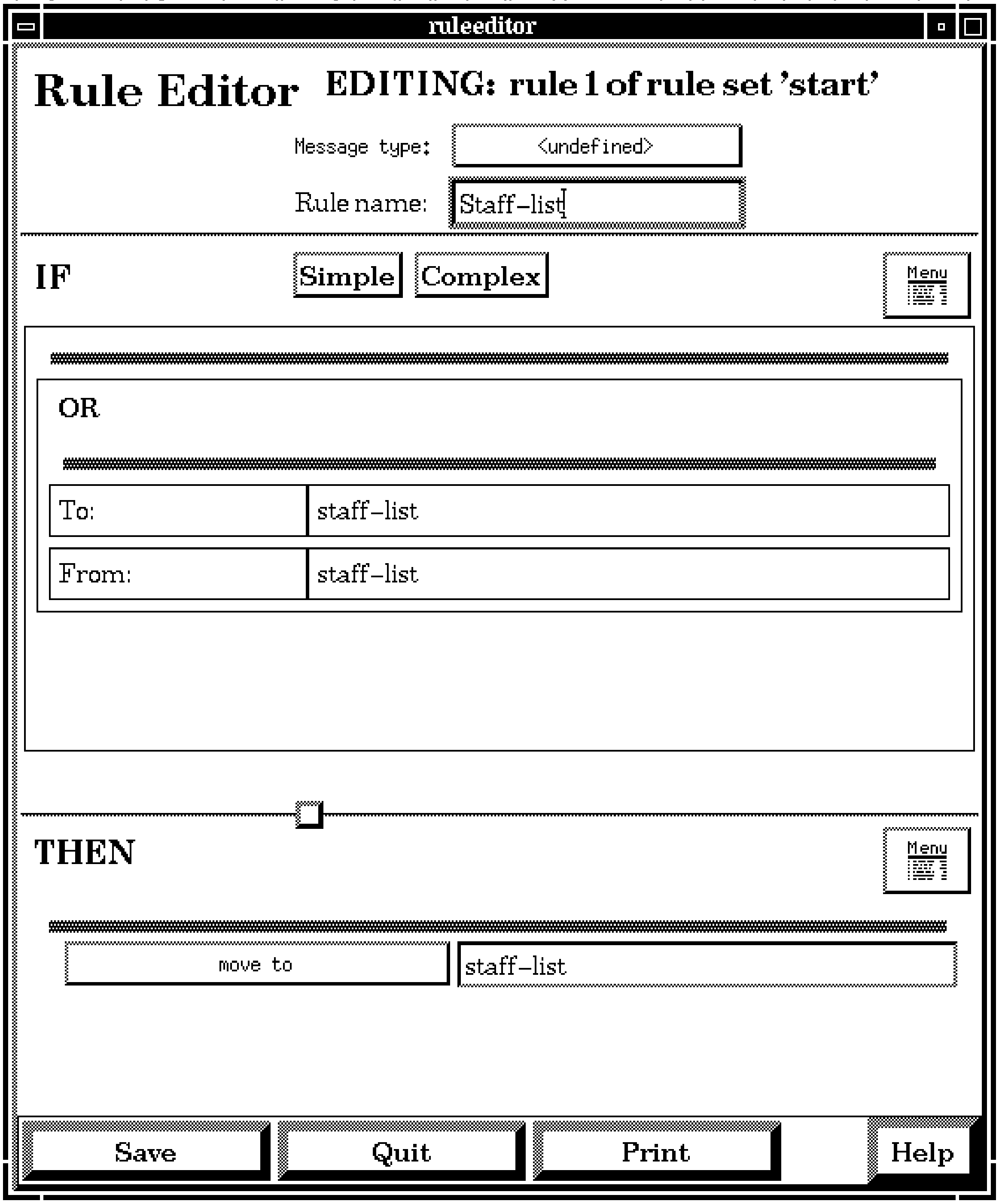

Figure 7: Refile messages to a distribution group

You may find it useful to save copies of messages that you've sent. It may also be useful to use them to remind you of items that are still pending. One way to do this is to compose the message with an extra field (such as "Reminder: <date>), which you can then use later. The following rule will look for messages from you with a Reminder field, and display the message on the screen if it is after a particular date.

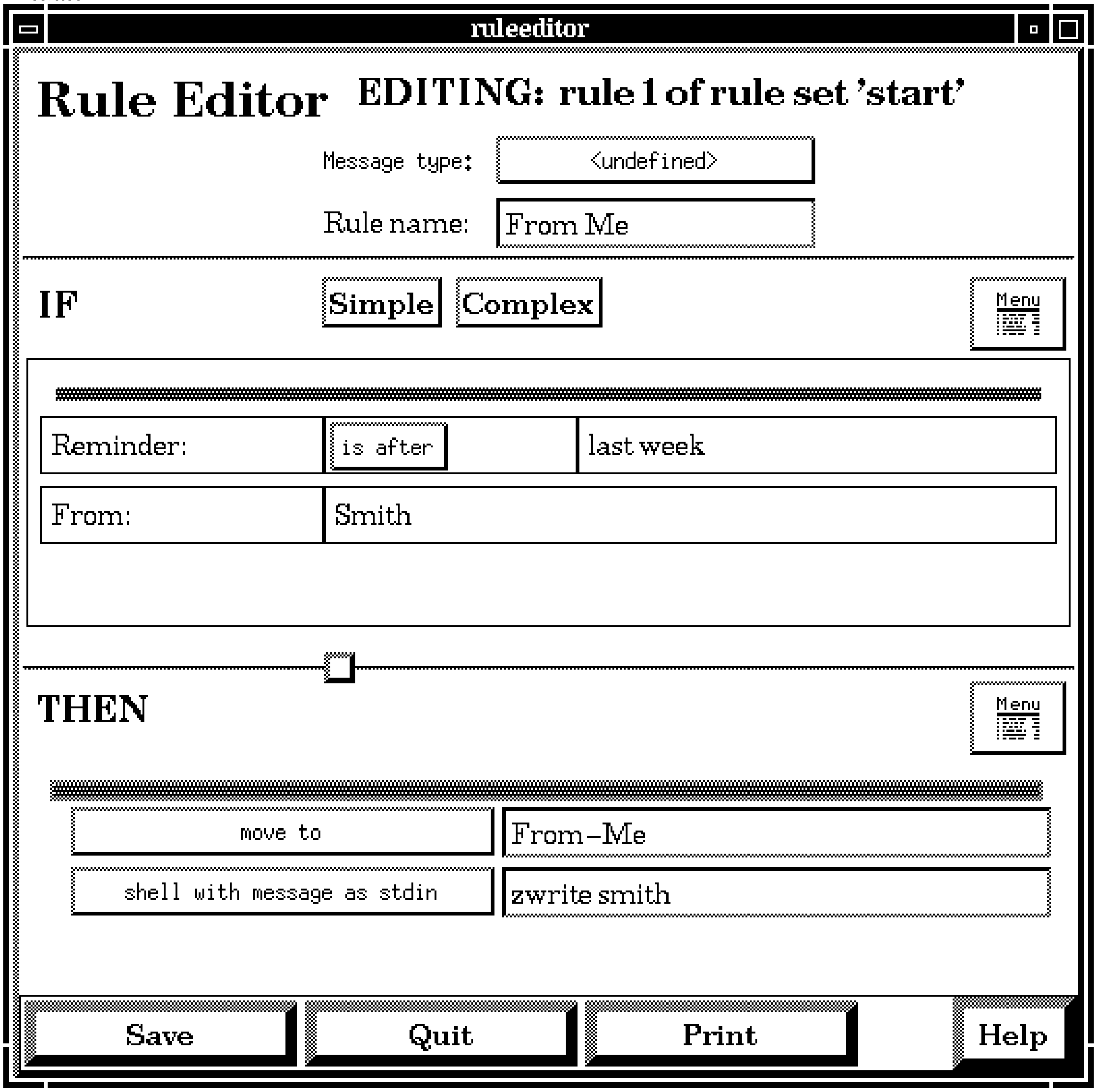

Figure 8: Remind me about messages I've sent

You can identify messages that are more than a certain length and print them.

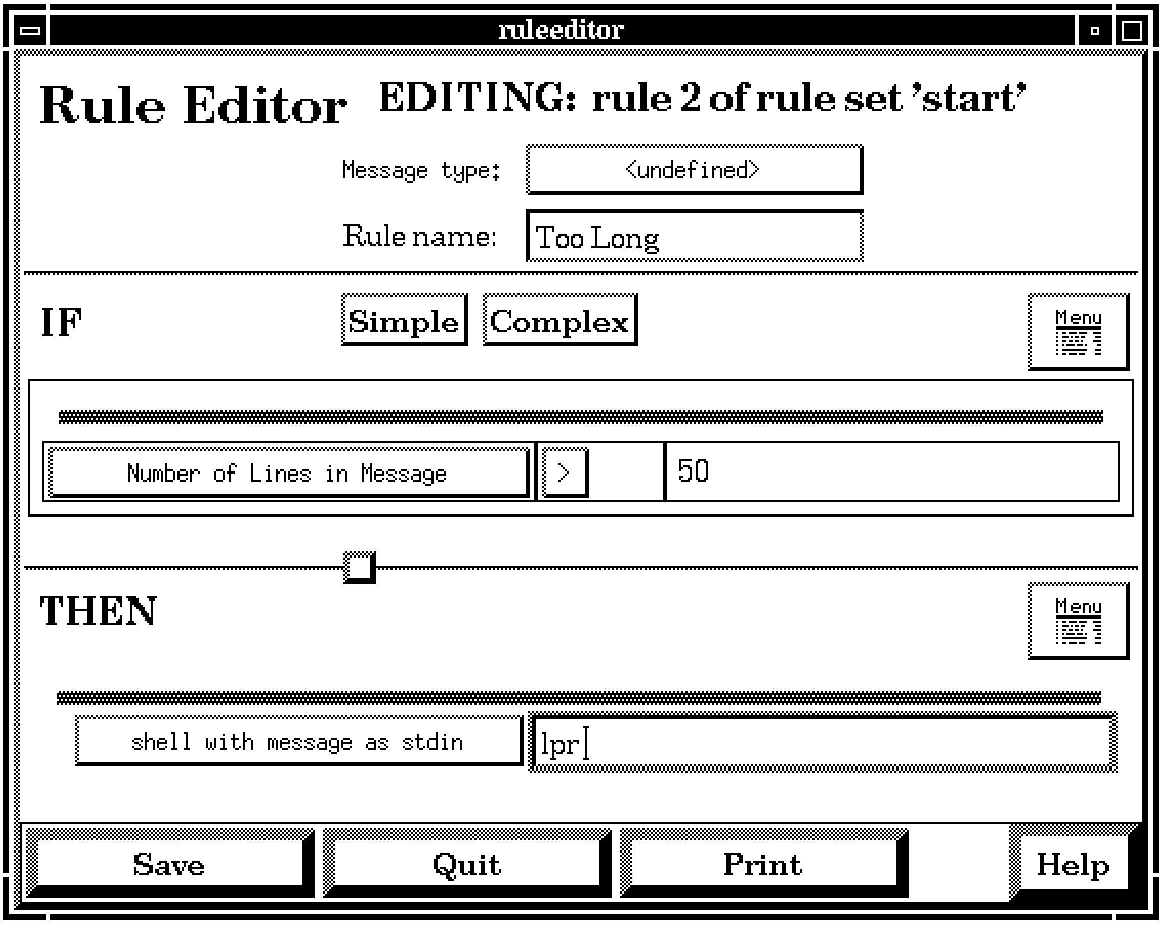

Figure 9: Print long messages

Members of a group may agree to use a particular field to help them organize a time for a meeting. The "meeting" field identifies the type of meeting. If this fires, it can call another ruleset, evaluate meeting, which processes the meeting times.

A person might send a message with the following fields:

Meeting: Argus Review
Monday: 1-5
Tuesday: False
Wednesday: True

The first rule would identify this type of meeting:

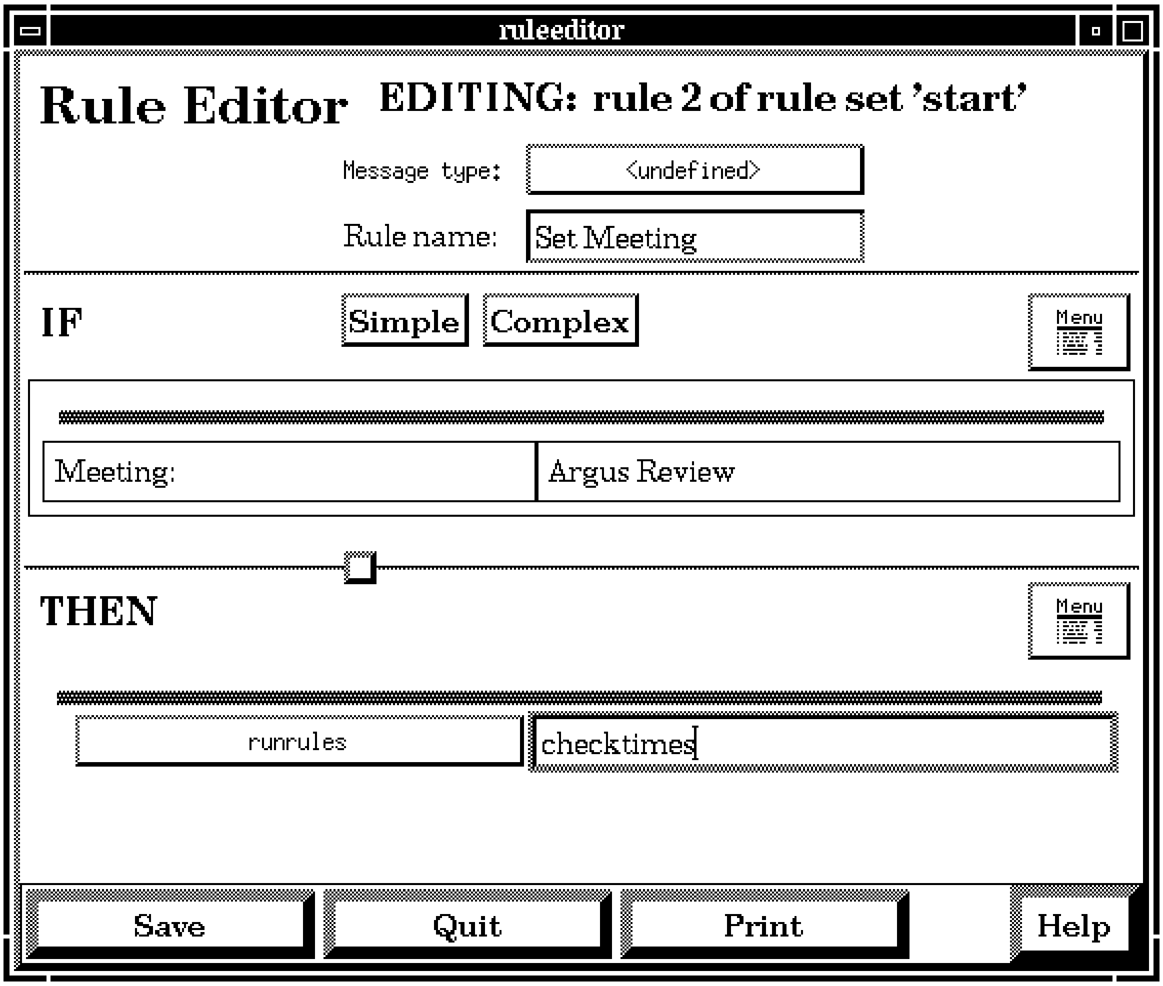

Figure 10: Organize a meeting

The second ruleset would contain rules that compile all of the possible times. Here is an example:

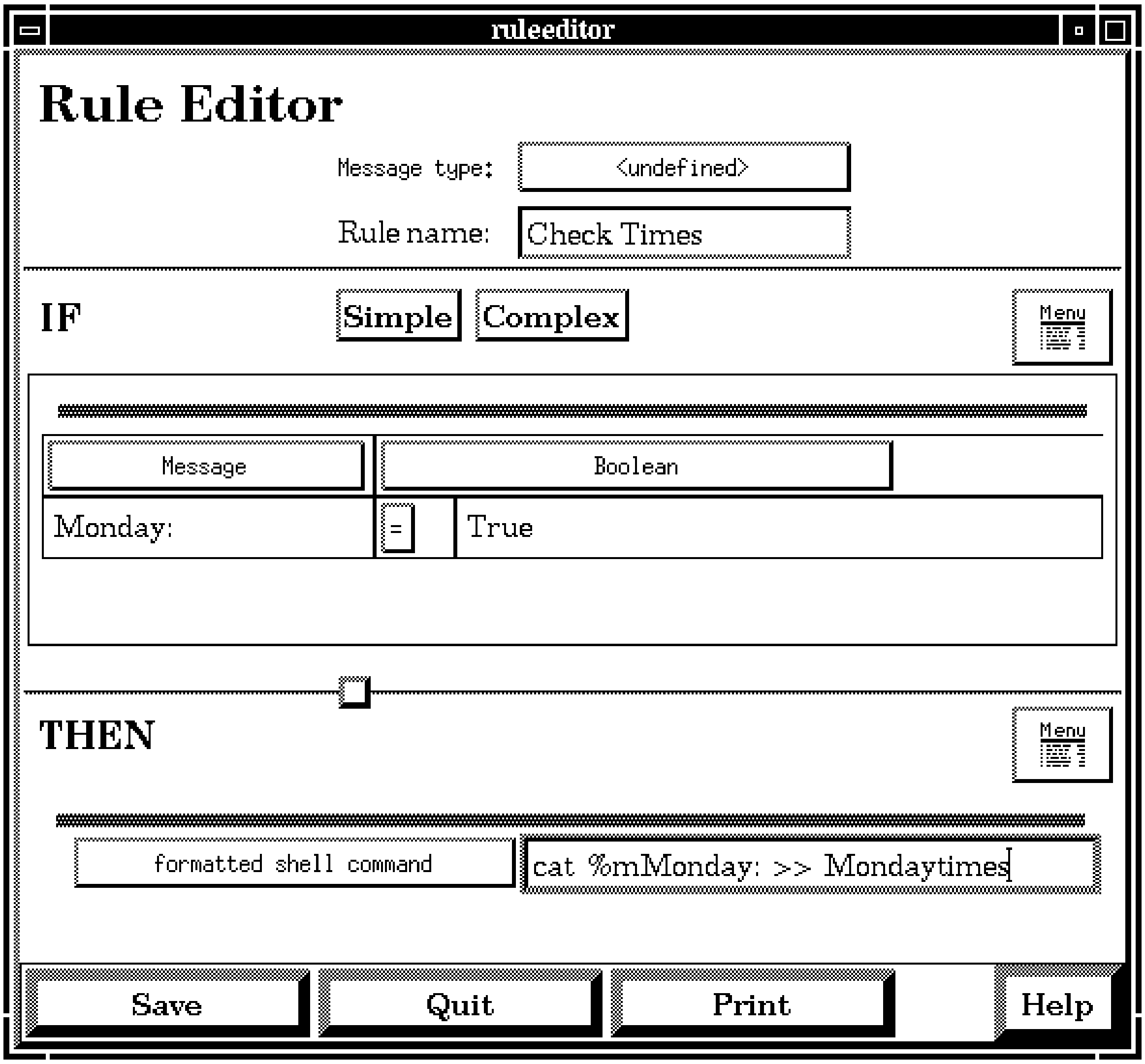

Figure 11: Check meeting times

You can use one rule to flag a message as important with a user property:

Figure 12: Set reminder property for important messages

You can then identify such messages with another rule:

Figure 13: Display messages with reminder property set

As you read your mail with mh (or Xmh), you can annotate the message with the "anno" command. This adds a field (with any name you specify) followed by a date.

For example, if you type:

```
anno -component 2-week-remind:
```

the following field will be added to the current message:

```
2-week-remind: Mon, 12 Feb 90 18:44:03 EST
```

Note: you can create aliases for commonly used "anno" commands, so that you do not have to type the whole string each time.

For example:

```
alias r2 'anno -component 2-week-remind:'
```

You can then add the following rule to your ruleset, to identify annotated messages after they are two weeks old:

Figure 14: Remind me about this message if it is more than 2 weeks old

You can also use the "anno" command to create messages that "self-destruct" after a period of time. For example, messages from people announcing that they will be away on vacation for the next week.

```
alias d2 'anno -component 2-week-delete:'
```

Figure 15: Delete this message if it is more than 2 weeks old

You can check if a message has already been moved or deleted, and only process it if it has not.

Figure 16: Only move it this message hasn't been processed yet

Table of Contents

List of Figures