Zephyr
on Athena

Sharon Belville
Athena Documentation

10 September 1991
Version 3

COPYRIGHT NOTICE

TRADEMARK NOTICES

REVISION HISTORY

Version 3 (10 Sep 1991, Sharon Belville): Updated to 7.3, generally revised. Thanks to Jeanne Cavanaugh and Phyllis Galt.

Version 2 (29 Jan 1991, Mark W. Moormans): Updated cover material to include copyright and trademark notices.

Version A (2 Feb 1990, Sharon Belville): Original document.

Overview

Zephyr is Athena's notice transport and delivery system. In English, Zephyr is a way for users and machines to send real-time (nearly instantaneous) messages to each other.

Your most common interaction with Zephyr will be with the Zephyr message windows (called "windowgrams") that will occasionally appear on your screen. You can get rid of a windowgram by clicking on it with any mouse button. Electronic communication with Zephyr is only possible when both sender and recipient have activated Zephyr on their workstations and are logged in simultaneously. The primary advantage of using Zephyr over electronic mail is that it is much faster and messages are more immediately noticeable to the recipient.

With several hundred workstations scattered across the MIT campus, Zephyr is the way to find someone or send a quick message to a person or a group of people. Its ability to centralize information and distribute it quickly makes it useful for tasks like user login detection, wide distribution of time-critical information, and sending easily disposable (or filterable) messages instantaneously.

This document describes the following aspects of Zephyr:

• **Using Zephyr** describes basic Zephyr techniques, such as sending windowgrams, locating other users, receiving automatic mail notification via Zephyr, etc.

• **Customizing Zephyr** describes how to tailor your Zephyr environment with regard to fonts, windowgram location, the Zephyr description file `~/.zwgc.desc`, etc.

• **Troubleshooting Zephyr** notes some errors you might encounter using Zephyr, and how you can deal with them.

• **Other Sources of Information about Zephyr** summarizes other information resources to help you learn about Zephyr.

Using Zephyr

This section covers the basics of using Zephyr:

- how Zephyr is automatically alerted when you log in
- using the **zwrite** program to send messages to one or more users
- using **zlocate** and **znol** to locate a user or a list of users
- what a Zephyr subscription is, what's available, and how to subscribe
- setting whether other users can locate you
- sending automatic messages, either to yourself or others
- having Zephyr notify you whenever new electronic mail arrives for you
- using Zephyr over dialup (i.e., from a non-X terminal)
- saving Zephyr notices to a file

How Zephyr Starts Up (zwgc)

The primary program that Zephyr uses is **zwgc**, or the Zephyr WindowGram Client, which is responsible for displaying Zephyr notices. By default, the **zwgc** program is *automatically* started for you at login and activates Zephyr on your workstation. When you log in to Athena, the following Zephyr initializations occur:

- You are automatically set up to receive personal messages, as well as messages from Athena operations and messages related to appropriate fileservers.

- You become *locatable*, meaning other Athena users can find out where you are logged on, and what time you logged on there.

Sending Messages to Other Users (zwrite)

Zwrite to a Specific User

The simplest and most commonly-used command of Zephyr is **zwrite**. This command sends a message from you to another user, as long as you are both logged in. To **zwrite** to someone, type:

zwrite *username*

Where *username* is the username of the person to whom the message is to be sent. If that person is not currently logged in, **zwrite** responds:

```
username:  Not logged in or not subscribing  to messages
```

Otherwise **zwrite** responds:

```
Type your message now.  End with control-D or a dot on a line by
itself.
```

Now type in your message, ending it by pressing **Return** then entering **Ctrl-d** or a period on a separate line. Be sure to hit the **Return** key at the end of each line: **zwrite** does not wrap long lines, so they could extend past the edge of the recipient's screen. There is no limit to the length of the messages you can send with **zwrite**, although messages are difficult to scroll through so you will want to create a message that will fit on one screen. Sometimes it takes a few seconds after you hit **Ctrl-d** or a period before Zephyr actually finishes sending the message. When the message is successfully sent, Zephyr returns the following confirmation:

```
username: Message sent
```

If you want to **zwrite** the same message to a number of different users, just include their usernames, separated by spaces, on the **zwrite** command line:

```
athena% zwrite  abbott  costello
```

Broadcasting a Zephyr Message to Subscribers

Users can specify certain categories of messages they would like to receive by *subscribing* to them. These categories are called *instances*. The idea of Zephyr subscriptions is covered in the section **Subscribing to Zephyr Messages**.

To send a message to a specific instance, use the **-i** option with **zwrite**. For example, the following command sends your message to everyone who is subscribing to white-magic messages:

```
athena%  zwrite  -i  white-magic
```

You don't need to know specifically who is logged on (in fact, there's no way to tell who will be able to read it). There is no way to have a private instance, like a private mailing list, such that you can control who can read the messages.

Zephyr notices are also divided into *classes*. This is a more basic division than instances. You will rarely send messages to a class other than `message` (the default), but if you do, use the **-c** option to **zwrite**:

zwrite -c *classname*

As with class `message`, if you do not specify an instance when you specify a class, **zwrite** will default to instance `personal`. You can use **-c** and **-i** together on the **zwrite** command line.

There are many other options available with **zwrite**. See the man page for more information.

Formatting Text in Zephyr Messages

Zephyr lets you include formatting commands in the text of your messages to control how the text will appear on the recipient's screen. (The recipient can override these formatting commands; see the section on the ~/.zwgc.desc file.)

A formatting command can be embedded anywhere in your text, and must be of the form:

@command(text)

Here *command* is the name of a Zephyr formatting command and *text* is the text to be formatted. You can use other pairs of delimiters than parenthesis to enclose the text to be formatted, including (), {}, [], and <>.

These following formatting commands are recognized by Zephyr:

```
--------------------------------------------
Command          Action
--------------------------------------------
@roman           nullify @italic and @bold
@b or @bold      boldface
@i or @italic    italics
@l or @left      left aligned
@c or @center    center aligned
@r or @right     right aligned
@small           small type size
@medium          medium type size (default)
@large           large type size
@huge            larger than large type size
--------------------------------------------
```

The following commands do not take text (to be formatted) as their arguments. The **@beep** command takes nothing (but still needs the delimiters), **@font** and **@color** take the relevant specifications.

```
---------------------------------------------------------------
@beep()             ring the X bell once
@font(fontname)     set the current font (e.g., courier, times,
                    or helvetica; use "xlsfonts | more" to list
                    other possibilities)
@color(colorname)   set the color (see /usr/lib/rgb.txt for
                    a list of possibilities)
---------------------------------------------------------------
```

Don't confuse the Zephyr formatting commands with commands you are used to from text-formatting programs such as Scribe (for example, **@begin(center)** and **@end(center)** do not work).

For example, try something like:

```
athena% zwrite abbott
Type your message now.  End with control-D or a dot on a line by
itself.
I'm asking @i[@b<you>] who's on first!
.
abbott: Message sent

athena% zwrite costello
Type your message now.  End with control-D or a dot on a line by
itself.
@large[That's the man's name.]
.
costello: Message sent
```

Finding Other Users (zlocate and znol)

Finding a Specific User

Zephyr lets you locate other users if they are logged on and using Zephyr during your login session. To locate someone whose Athena name is *username*, type:

zlocate *username*

If that user is logged in and is not "hidden", **zlocate** responds by telling you where and when the user is logged in. For example:

```
athena% zlocate costello
M66-080-17.MIT.EDU               unix:0  Mon Nov 27 15:26:43 1989
```

It is possible to be logged in and be hidden. If the user you want to **zlocate** is either not logged in or not "exposed" to Zephyr, **zlocate** returns:

```
Hidden or not logged-in
```

To learn how to hide from Zephyr, see **Setting Zephyr Exposure**.

Finding Several Users

If you want to locate multiple users, you can add their names to the original **zlocate** command line separated by spaces.

```
athena%  zlocate  abbott  costello
```

The command **znol** provides two services. When first run, the **znol** command will go through a list of usernames and **zlocate** each one. Thereafter it will notify you by windowgram when people in this same list log in or out. This name list is usually in a file ~/.anyone in your home directory. To execute **znol**, create a file called ~/.anyone, and enter a list of usernames, each on a separate line, such as:

```
abbott
costello
hardy
laurel
```

Then type:

```
athena% znol
```

For each person in your ~/.anyone file, **znol** lists where and when each person is logged in. For example:

```
abbott: M14S-010-1.MIT.EDU       unix:0  Mon Nov 27 16:05:24 1989
costello: M66-080-17.MIT.EDU     unix:0  Mon Nov 27 15:26:43 1989
laurel: M11-113-14.MIT.EDU       unix:0  Mon Nov 27 12:35:29 1989
```

As **znol** tries to locate each user, it subscribes you to the login messages from that user. So, if that user is announcing his or her logins, you will get windowgrams as that user logs in and logs out. See **Setting Zephyr Exposure** and **Subscribing to Zephyr Messages** for more information.

You can specify a file other than ~/.anyone by using the **-f** option with **znol**:

znol -f *filename*

If you decide you do not want to receive a windowgram every time a person logs in or out, you can disable **znol** for the rest of your session:

```
athena% znol off
```

For information on other options to **znol**, type **man znol** at your `athena%` prompt.

Subscribing to Zephyr Messages (zctl)

Introduction to Zephyr Subscriptions

To receive a certain type of electronic notification from Zephyr, you must first be subscribed to that type. Subscriptions work very much like real magazine subscriptions; you get to decide which ones you want to be delivered to your workstation. Every Zephyr subscription consists of three specifications:

- class - describes the general subject of the message
- instance - subdivision of class
- recipient - the username the particular kind of message is sent to

Most Zephyr notifications that will concern you are class "`message`", with the instance describing the type of message. The recipient field defaults to your username, although you may also subscribe to recipient "`*`", that is, everyone subscribed to that class and instance. A typical subscription (as returned by the **zctl** program) might look like this:

```
Class message instance personal recipient abbott@ATHENA.MIT.EDU
```

indicating that user `abbott` is subscribed to personal messages addressed to him.

Subscriptions are handled through the **zctl** program, which is short for Zephyr control. You use **zctl** to subscribe to different notice types and then to save the subscriptions in a file (by default the file ~/.zephyr.subs). Every Athena user is automatically subscribed to two subscriptions:

```
Class message instance personal recipient username@ATHENA.MIT.EDU
Class operations instance message recipient *
```

If you only have one or two simple changes to make to your subscriptions, you can use command line options on the initial **zctl** line without actually entering the **zctl** program. For instance, if you want to subscribe to messages that are class `message` and instance `urgent`, use the **sub** option or **add** option. The **sub** option subscribes you temporarily (for the duration of your login):

```
athena% zctl sub message urgent
```

The **add** option adds the new subscription to your ~/.zephyr.subs file so that you will be subscribed whenever you log in:

```
athena% zctl add message urgent
```

To enter the **zctl** program to add a number of subscriptions to your ~/.zephyr.subs file, type the command with no options:

```
athena% zctl
ZCTL $Revision: 1.19 $ (Protocol ZEPH0.2) - Type '?' for a list of
commands.

zctl:
```

You are now ready to execute **zctl** commands. To list your current subscriptions, use the **retrieve** command while in **zctl**. (Note that it will probably return several subscriptions besides the two defaults. When you attach filesystems, you become subscribed to operations messages concerning the filesystem.)

The **zctl** command **?** or **lr** will now list all the available commands with a brief summary of each.

Some Zephyr Message Classes and Instances

This section describes some of the interesting classes and instances to which you may subscribe.

Class

The `message` class is the most frequently used; in fact, **zwrite** defaults to class `message` if you do not specify a class. The common classes are shown in this table:

```
-----------------------------------------------------------------
CLASS       INSTANCE                 SYNOPSIS
-----------------------------------------------------------------
```

Message	(See list below)	default class
Login	**username@athena.mit.edu**	Login and logout notices
Filsrv	**machinename**	Shutdown messages
Filsrv	**machinename:hesiodname**	Filesystem messages
Discuss	**machinename:path**	Discuss transaction notices
Mail	**pop** and **popret**	New mail notification
Operations	**message**	Operations messages

Zephyr users are automatically subscribed to `Operations` class notices, appropriate `filsrv` class notices (according to machines that serve the workstation and the appropriate filesystems), and class `message`, instance `personal` messages. Running **znol** (See **Finding Other Users**) will automatically subscribe (for the login session) to class `login` messages for each specified user.

Instance

Every Zephyr user is automatically subscribed to `personal` instance messages, which is the default message from one user to another. The different instances are not registered anywhere, so creating a new instance is as simple as telling users to subscribe to that instance (recipient "*"), then sending messages to that instance. This means that it is impossible to compile a complete list of instances.

The easiest way to find out interesting instances is to subscribe to class `message`, instance `*`, recipient `*`, which subscribes you to every non-personal message which comes across the network (note that you cannot subscribe to class `*`):

```
athena% zctl sub message  \*  \*
```

Subscribing to "star star" (as it's colloquially known) can get noisy and distracting when you're working and lots of windowgrams appear on your screen, but you will be able to decide which instances are worth your attention. You can learn of others (that may not necessarily appear while you are logged in) through word of mouth.

Here is a partial list of instances belonging to class `message`. Note that none of these instances are officially sanctioned.

INSTANCE	SYNOPSIS
personal	User to user messages.
urgent	Urgent user messages.
proverb	Proverb every hour.
weather	Hourly weather report (on the half hour).
xconq	Messages concerning xconq, especially finding opponents.
boggle	Automatic message when someone starts playing

```
               mboggle.
  hunt         Automatic message when someone starts playing hunt.
               Both boggle and hunt can also be used to find
               opponents.
  consult      Consultant messages for consultant communication;
               Don't ask questions over instance consult - use olc.
               Consultant use of this instance is subject to
               change.
  help         Informal questions other users can answer.
  sipb         Questions, answers, and discussions involving SIPB
               (Student Information Processing Board) members.
  watchmaker   Technical discussions about Athena. Watchmakers are
               student programmers employed by Athena system
               development.
  apo          Discussions involving Alpha Phi Omega members.
  white-magic  Random discussions, hard to define.  Listen in if
               you're not too busy.  A voluminous instance.
               Read /mit/belville/WM.NOM.
  e40-lnf      Discussion about Late Night Food, from prices to
               invitations.
  -----------------------------------------------------------------
```

Some living groups have an instance, for example, `fenway` (Fenway House) , `chiphi` (Chi Phi), `nh2` (New House II), `Desmond` (New House V), `tdchi` (Theta Delta Chi), and `nud` (Nu Delta).

Some courses may also have an instance, especially if the course uses Athena. For example, `1.00` and `6.170`. This is a good way to discuss assignments, quizzes, etc.

Setting Zephyr Exposure (zctl)

Zephyr allows you various exposure levels; in other words, you control whether or not people can **zlocate** you. To set an exposure level you need to set a variable from within **zctl**:

zctl set exposure *exposure-level*

where *exposure-level* can be one of the following (listed from widest exposure to least exposure):

```
-------------------------------------------------------------------
net-announced       You are registered with Zephyr.  Anyone can
                    locate you or receive your login notices.
                    Default subscriptions and any additional
                    personal subscriptions will be entered.

net-visible         You are registered with Zephyr.  Anyone can
                    locate you.  Only users within the same Kerberos
                    realm (i.e., @athena) can receive login notices
```

```
                      about you.  (Type man kerberos for more
                      information.)  Default subscriptions and any
                      additional personal subscriptions will be
                      entered.

realm-announced       You are registered with Zephyr.  Anyone within
                      the same Kerberos realm can locate you or
                      receive login notices.  Default subscriptions
                      and any additional personal subscriptions will
                      be entered.

realm-visible         (default) You are registered with Zephyr.
                      Anyone within the same Kerberos realm can
                      locate you.  No login or logout announcements
                      will be sent.  This is the system default.
                      Default subscriptions and any additional
                      personal subscriptions will be entered.

opstaff               You are registered with Zephyr.  Only Athena
                      operations staff can locate you, and no login
                      notices will be sent.  Default subscriptions
                      and any additional personal subscriptions
                      will be entered.

none                  This completely disables Zephyr for the user.
                      You are not registered with Zephyr.  No one can
                      locate you, no login messages will be sent.  No
                      subscriptions will be entered, nor any notices
                      displayed.
-----------------------------------------------------------------
```

The Kerberos realm for users with Athena accounts is @athena.mit.edu. Type **man kerberos** for more information.

You have the option of hiding during your login session, but still receiving Zephyr notices, the equivalent of exposure level "opstaff". To do this type:

```
athena% zctl hide
```

To become visible again (set the exposure back to normal), type:

```
athena% zctl unhide
```

Sending Automatic Messages (zaway and zleave)

There are two kinds of Zephyr messages you might want to send automatically:

- to other users, e.g. to tell them you can't respond right now
- to yourself, as reminders

Sending Automatic Replies to Others (zaway)

To send an automatic reply to users who send you a message via **zwrite**, use the **zaway** program. Type:

```
athena% zaway
```

with no **&**. This is typically run when you leave your workstation temporarily. The **zaway** program uses a message file (default is ~/.away) to specify messages for users.

If you run **zaway** and no ~/.away file exists, and no other file is specified, the following message is returned:

```
I'm sorry, but I am currently away from the terminal and am
not able to receive your message.
```

The ~/.away file allows you to specify different messages for different users. The general format of this file is:

```
>username
message
>username
>username
message
```

There are two special names:

- "%" specifies users that have not been explicitly named yet in the file

- "*" indicates the message should be sent to all users.

If a user name does not match any of those listed in the file, and no "*" or "%" field is specified, no return message is sent. All messages are preceded by the line `Automated reply:`.

Here is an example of a ~/.away file:

```
>abbott
>costello
Who's on first, what's on second, and
>%
I'm sorry, I'm not able to respond right now,
>*
I'll be back in a few minutes.
```

The last line will be appended to all the above messages. Abbott and Costello will get the `who's on first` line, and everyone else will get the `I'm sorry` line. Everyone will get the last line, `I'll be back`.

You can specify a file other than ~/.away by typing:

zaway *filename*

To exit from **zaway**, use **Ctrl-C**. Note that you can only have one **zaway** running on a workstation.

Sending Automatic Messages to Yourself (zleave)

The **zleave** program sets an "alarm". You supply the time you want to leave, then Zephyr sends you reminders 5 minutes before, 1 minute before, at the actual time, and every minute for ten minutes thereafter (or until you log out or cancel the **zleave** process).

You can set the time as a specific time (within the next 12 hours) using the format *hhmm*:

```
athena% zleave 1145
Alarm set for Mon Jan  8 11:45:50 1990
```

or as a relative time (as in +20 minutes from the current time):

```
athena% zleave +0005
Alarm set for Mon Jan  8 11:33:20 1990
```

The default messages are:

```
Message                           When Sent
----------------------------------------------------------------
You have to leave in 5 minutes    5 minutes before specified time
Just one more minute!             1 minute before
Time to leave!                    Specified time
You're going to be late!          Each minute following
----------------------------------------------------------------
```

If you want to add to the message, use the **-m** option to **zleave**:

```
athena% zleave +0020 -m "Are you done yet?"
```

Your message will append to the default message, as in:

```
Just one more minute!
Are you done yet?
```

You might simply want to add a beep:

```
athena% zleave +0200 -m "@beep()"
```

To cancel the **zleave** messages, type:

```
athena% zleave cancel
```

or

```
athena% zleave can
```

The Zephyr messages are sent over instance leave (recipient your username). When you start **zleave**, you are subscribed to this instance, but cancelling **zleave** does not unsubscribe you. This should not be a major concern, as you only receive messages specific to you.

Getting Zephyr Notification of New Mail

To receive simple notifications of new mail, subscribe to class `MAIL`, instance `POP` messages. You do this by typing:

```
athena% zctl add mail pop
```

to your prompt. After doing this, you will get a simple notice every time you are logged in and more mail arrives for you at your post office.

If you wish to be notified of the sender, recipient and subject of the new mail, you must also remove the "#" signs from the ten lines from `match MAIL` to `exit`, in your ~/.zwgc.desc file, and subscribe to class `MAIL`, instance `POPRET` messages:

```
athena% zctl add mail popret
```

If you don’t have a ~/.zwgc.desc file, you can get one with the command:

```
athena% cp  /usr/athena/lib/zephyr/zwgc.desc  ~/.zwgc.desc
```

This will destroy any ~/.zwgc.desc file you have, so only do this if you don’t have one.

Note: Uncommenting those lines and subscribing to class `mail` instance `popret` is *not* necessary to receive notifications of new mail. The only effect of uncommenting these lines is to display on your screen the sender, recipient, and subject of the mail, rather than just a message informing you of new mail. In addition, uncommenting these lines will add extra load to the post office servers, making them run slower.

Using Zephyr Over Dialup

If you log into Athena by dialing in over a modem, you are probably on a terminal that cannot handle windowgrams. If you still want to get Zephyr messages, you need to tell Zephyr you are in a "tty" session, and the messages will appear as regular text on your screen.

To start Zephyr in tty mode whenever you are logged in via dialup, you need to type this line once:

```
athena% zctl set fallback true
```

You can also start **zwgc** in tty mode. Be sure to kill a previous **zwgc**. Follow this procedure to find and kill other **zwgc**'s, then start up the new one:

```
athena% ps ux | grep zwgc
abbott  24722  0.0  4.5  369  280 v0 I     0:06 /usr/etc/zwgc
athena% kill 24722
athena% zwgc -ttymode
```

This will do the same thing as above - displaying your messages as regular text on the screen. NOTE: Never start **zwgc** with an **&**, or the program will not exit correctly when you log out.

If you don't want the messages appearing, but you want to read them at your leisure, see **Saving Zephyr Notices to a File**.

Saving Zephyr Notices to a File

Saving the Text of a Zephyr Message

You can save information from a Zephyr windowgram by using cut-and-paste. Since clicking any mouse button on a windowgram will cause it to disappear, you must use the **Shift** key before any mouse command. The following commands will cause the text to be highlighted and put into the X cut buffer. This text can be pasted into another (non-windowgram) window, such as an xterm or an Emacs window:

Shift-Click **LEFT** button	Mark one end of the region to be copied
Shift-Click **RIGHT** button	Mark other end of the region to be copied and highlight
Shift-Drag **LEFT** button	"Dragged" text will highlight
Shift-Drag **RIGHT** button	Extend already highlighted text

There is no way to cut-and-paste "special effects" such as boldface, italics, or large print. This special text will get pasted as normal text.

Saving All Incoming Zephyr Notices to a File

Sometimes when you're busy or logged in via dialup, you may want to save your messages to a file, so that you are not continually bombarded with messages but you can still receive them and read them at your leisure.

If you want to choose when you save the messages to a file, rather than automatically doing it every time, you can run a command of the following form from the `athena%` prompt:

zwgc -ttymode > /tmp/*filename*

This will save incoming Zephyr messages in a file in the /tmp directory called *filename*. (It is a good idea to create this file in the /tmp directory, so that it doesn't count against your quota; but note that files in the /tmp directory will be deleted when the workstation deactivates.)

To see the new messages as they are written to the file, run:

tail -f /tmp/*filename* &

If you lose a message, you can look at the file to see what you missed. If you don't run **tail**, you can just read the file whenever you want, as long as it is within that login session. You can copy that file to your directory if you want to save them for a longer period.

If you want to save messages to a file automatically when you log in via dialup, you need to do the following to prevent an automatic **zwgc** starting:

`athena%` **`zctl set fallback false`**

Then put the following line into your ~/.startup.tty file:

zwgc -ttymode > /tmp/*filename*

Customizing Zephyr

This section covers more advanced Zephyr concepts. If you are not very comfortable with Zephyr or with using the different Zephyr-related files, proceed with caution. If you do not understand something, don't panic, just ask for help. There are many people around Athena who are well-versed in the concepts covered here.

Many customizations for Zephyr windowgrams involve the ~/.Xresources file, which is referred to by the X window system. This section refers to the ~/.Xresources file, but does not cover it in detail. For more information on X manipulation, type the following commands at your `athena%` prompt:

```
athena% add X
athena% man X
```

You can control many aspects of the windowgrams you receive (e.g., placement on your screen, use of fonts, etc.) by using the description file ~/.zwgc.desc. The language is fully described in the manual page for **zwgc** (type **man zwgc** at the `athena%` prompt). The default ~/.zwgc.desc file is found in /usr/athena/lib/zephyr/zwgc.desc. The best way to learn the new language is to copy this file to your home directory (~/.zwgc.desc), then you can make changes and experiment.

After making changes to your ~/.zwgc.desc file, you can incorporate them into your current Zephyr session by typing the following at your `athena%` prompt:

zctl wg_read

Otherwise the changes will take place next time you log in.

There is also a directory in the zephyr locker (/mit/zephyr/examples) that contains code fragments for ~/.zwgc.desc files. You can look here for ideas. If you come up with a useful example of how to massage the ~/.zwgc.desc file, send it via electronic mail (with descriptive comments such as those found in the current examples) to zephyr-comments@athena.

This section covers the following topics, including examples for the ~/.zwgc.desc file and the ~/.Xresources file, as well as other customizing abilities:

- changing the signature that appears in windowgrams you send

- sending Zephyr notices in different fonts, and changing the fonts that your received windowgrams appear in

- using your ~/.zwgc.desc file to filter out unwanted Zephyr notices

- using the ~/.zwgc.desc file to format Zephyr notices
- using the ~/.Xresources file to position Zephyr notices on your screen

Setting Your Zephyr Signature

When you send a Zephyr message to someone, your *signature* appears in the message. If you do not explicitly set it to something else, it will be your name as it appears in your **finger** information.

There are two ways to set this signature. You can set it permanently:

```
athena% zctl set zwrite-signature "Lou Costello"
```

Or, you can set it for a specific Zephyr message:

```
athena% zwrite abbott -s "Lou Costello"
```

This **-s** option will override the signature if previously set in either your **finger** information or your **zwrite-signature** variable.

If you don't want any signature, set an empty signature:

```
athena% zwrite abbott -s ""
```

Zephyr Fonts

Changing the Font in Messages You Send

There are 3 fonts defined for you in the standard **zwgc**. They are `courier` (the default), `helvetica`, and `times`. You can send someone a message in `times` by putting the **@times** command in the text of the message. For example, in the following Zephyr message, the second line will appear in times font:

```
athena% zwrite abbott
Type your message now.  End with control-D or a dot on a line by
itself.
the next line will be in "times" font.
@times(this line is in "times".)
this line is back to the default font.
```

You can also use any of the fonts available for X, using the **@font** command.

```
athena% zwrite costello
Type your message now.  End with control-D or a dot on a line by
itself.
```

```
@font(9x15)
This is printed in the font named "9x15"
```

You can bind the scope of a font change by adding a blank environment like so:

```
athena% zwrite costello
Type your message now.  End with control-D or a dot on a line by
itself.
@(@font<6x10>  Text in 6x10 font)  Default
```

You can get the list of X fonts by typing the following:

```
athena% xlsfonts | more
```

Anything that is not specified in a different font will use the default font, which is set to courier. You can change the default font for messages you receive (but not for messages you send). See **Changing the Default Font in Zephyr** for instructions.

Defining Your Own Fonts

Each of the default font families (times, helvetica, and courier) has 4 fontfaces associated with it: `bold`, `italic`, `bolditalic`, and `roman`. (Roman means that there is no bold or italic applied to the font.)

Each also has 3 sizes associated with it: `small`, `medium`, and `large`. This allows you to put:

```
print "@large("+$sender+")"
```

in your ˜/.zwgc.desc file, for example, to print the sender's name in a larger font than normal. You can also send someone a **zwrite** message like this:

```
hi, this message has @large(large words) in it and
@small(small words) in it, as well as normal (medium)
sized words...
```

So, in order to "fully" define a font family for **zwgc**, you need to find twelve X fonts that will complete your font family (3 sizes times 4 faces), and insert their names into your ˜/.Xresources file. (You can find font names by typing **xlsfonts | more**.) For example, the standard `times` font family is defined like so:

```
! (This is taken from the file:  /usr/athena/lib/zephyr/zwgc_resources  )
!
! The following is the adobe-times font family.  Availiable sizes are
! 80, 100, 120, 140, 180, and 240.  This family used to be times-roman.
!
*fontfamily.times.small.roman:          *adobe-times-medium-r-*-80-*-p-*
*fontfamily.times.small.bold:           *adobe-times-bold-r-*-80-*-p-*
*fontfamily.times.small.italic:         *adobe-times-medium-i-*-80-*-p-*
*fontfamily.times.small.bolditalic:     *adobe-times-bold-i-*-80-*-p-*
*fontfamily.times.medium.roman:         *adobe-times-medium-r-*-120-*-p-*
*fontfamily.times.medium.bold:          *adobe-times-bold-r-*-120-*-p-*
*fontfamily.times.medium.italic:        *adobe-times-medium-i-*-120-*-p-*
*fontfamily.times.medium.bolditalic:    *adobe-times-bold-i-*-120-*-p-*
*fontfamily.times.large.roman:          *adobe-times-medium-r-*-240-*-p-*
*fontfamily.times.large.bold:           *adobe-times-bold-r-*-240-*-p-*
*fontfamily.times.large.italic:         *adobe-times-medium-i-*-240-*-p-*
*fontfamily.times.large.bolditalic:     *adobe-times-bold-i-*-240-*-p-*
```

You could define your own font family named `foo` by replacing `times` with `foo` in the twelve lines above (in the columns on the left). Assuming you have defined a font family (let's use the name `foo` as an example), you can then put:

```
print "@foo("+$sender+")"
```

in your ~/.zwgc.desc file, for example, to print the sender's name in your newly defined `foo` font family.

If you don't want to have to find twelve fontfaces that work together, you can do just this:

```
zwgc.fontfamily.blah.Size.Face:     *adobe-times-bold-i-*-240-*-p-*
```

and then put:

```
print "@blah("+$sender+")"
```

in ~/.zwgc.desc to print out the sender in the `blah` font you defined. The problem with this, though, is that there is only one fontface and size defined, you can't use the formatting commands like **@b**, **@i**, etc. on it. Those commands would be ignored, and so something like:

```
@blah(@i(this) is @b(a) test)
```

would come out in just the `blah` font, without the bolding and italics done to it.

After you have changed your ~/.Xresources file, you can wait until the next login for the changes to take effect. If you want to see the changes during this login session, you must reload the X Resource DataBase manager by typing:

```
athena% xrdb ~/.Xresources
```

and then you'd have to kill and restart your **zwgc**.

Changes to the ~/.zwgc.desc file, on the other hand, can be read in as soon as you save out that file, by typing:

```
athena% zctl wg_read
```

Changing The Default Font In Zephyr

If you want to change the default font that is used when displaying zephyrgrams, you need to include the following line in your ~/.Xresources file:

```
zwgc*style*substyle.default.fontfamily:         times
```

This would use `times` as the default font. You can use one of the predefined fonts: `times`, `helvetica`, or `courier` (which is the Athena default). Or, if you have defined your own font families in your ~/.Xresources file, you can use one of those names.

Whenever you change your ~/.Xresources file, you must either wait until the next login to incorporate the changes, or reload the X Resource DataBase manager by typing:

xrdb ~/.Xresources

and then kill and restart your **zwgc** (or just wait until next time you log in) in order to see the new fonts.

Ignoring Zephyr Messages

If you want to permanently filter out a specific *class*,*instance*,*recipient* notice, even if you are subscribed to all messages via `message * *`, you should add a line to your ~/.zephyr.subs file that looks like this:

-*class*,*instance*,*recipient* For example, to ignore class `message`, instance "`annoy`" notices, you would insert this line:

```
-message,annoy,*
```

Note that although **zwgc** understands these "unsubscriptions", **zctl** does *not*. Therefore, if you do a **zctl load** or a **zctl wg_read** after adding them to your ~/.zephyr.subs file, they will not be filtered properly. You need to wait until the next login, or kill your **zwgc** and restart.

```
athena% ps aux | grep zwgc
costello 26266  2.5  3.9  337  244 p1 S     0:00  (zwgc)
athena% kill 26266
athena% zwgc
```

Formatting Zephyr Notices Using .zwgc.desc

Here is an example that uses variables and formatting commands to change the way a windowgram is displayed. In the code for displaying mail notification (from the ~/.zwgc.desc file mentioned above), change the line

```
print substitute($default)
```

to:

```
print "New mail from @bold("+protect($1)+"), Subject:
@italic("+protect($3)+")\n"
```

This changes the mail notification from:

```
(Authentication: UNAUTHENTIC)
You have new mail:

From: abbott@ATHENA.MIT.EDU
To: costello@ATHENA.MIT.EDU
Subject: Who's on first.
```

to:

```
(Authentication: UNAUTHENTIC)
```

```
New mail from abbott@ATHENA.MIT.EDU, Subject: Who's on first.
```

In Zephyr mail messages, variable 1 is the sender, variable 2 is the recipient, and variable 3 is the subject. The "\n" command puts a new line at the end of the message. The concepts of variables and commands are discussed in the **zwgc** man page.

Positioning Zephyr Notices on Your Screen

The recommended way to change where Zephyr notices appear on your screen uses the ~/.Xresources file. The geometry should be of the form *{+|-}x{+|-}y*, specifying an (x,y) coordinate for a corner of the window displaying the notice. The interpretation of positive and negative location specifications follows the X conventions. A special location of `c` for either x or y indicates that the window should be centered along that axis. For example, a geometry of `+0+c` specifies the window should be at the top of the screen, centered horizontally.

A typical section of your ~/.Xresources file might read like this:

```
zwgc.style.message.urgent*geometry:       +c+c
zwgc.style.message.personal*geometry:           -0-0
```

These lines would position `urgent` messages in the center of your screen, `personal` messages in the lower right corner, and by default, the others would appear in the upper left corner.

If you simply wanted all of your Zephyr notices to appear in the lower left, for example, you could use this line:

```
zwgc*geometry:          +0-0
```

Incorporating Changes to .zwgc.desc and .Xresources

Customizing Zephyr most often involves making changes to one or two files: ~/.Xresources and ~/.zwgc.desc. You can wait until the next login to see the effects of changes to either file, or you can incorporate the changes in your current login session.

To incorporate changes to ~/.zwgc.desc, you only have to type the following line:

```
athena%  zctl  wg_read
```

To incorporate changes to ~/.Xresources, you need to type the following line to make the changes to X:

```
athena%  xrdb  ~/.Xresources
```

and then kill your **zwgc** and start a new one to make the changes to Zephyr:

```
athena% ps aux | grep zwgc
costello 26266  2.5  3.9  337  244 p1 S 0:00  (zwgc)
athena% kill 26266
```

Troubleshooting Zephyr

Authentic and UNAUTHENTIC Zephyr Messages

When you receive a windowgram, the message is marked `Authentic` or `UNAUTHENTIC`. These labels indicate the result of the sender's or your own authentication.

To make a long story short, it is possible for another user to fool your workstation and make it believe he is somebody else. For this reason Athena has developed a system called Kerberos. Kerberos is able to verify that you are who you say you are and not impersonating someone else. Zephyr uses the Kerberos system to authenticate its users. (For more information, type **man kerberos** from the `athena%` prompt.)

If you and the sender passes authentication, then the message is marked with Authentic. If you *or* the sender failed authentication then the message is marked with UNAUTHENTIC. This does not necessarily mean that something bad is occurring. Any user can fail authentication if he loses his "Kerberos tickets" or his tickets expire (which they do after 10 hours). Any legitimate user can gain a new set of tickets and be automatically reauthenticated to the Zephyr server by running the **renew** program:

```
athena% renew
MIT Athena (w20-575-80)
Kerberos Initialization for "abbott"
Password:
```

If you are receiving all of your windowgrams marked UNAUTHENTIC, it is probably because you changed or renewed your Kerberos tickets. If you had recently typed **kinit**, then remember that you need to do **zctl load** as well. Zephyr will then receive the new tickets. The next time you need to use **kinit**, remember to repeat the above command.

Hostmanager Not Responding

Sometimes you may receive the following error messages when logging in:

```
main.c: Hostmanager not responding Setting Location
zctl (load): Hostmanager not responding: while subscribing
/bin/athena/zinit: Hostmanager not responding while subscribing
```

They may be mixed with other normal status messages. The common theme here is that your hostmanager is not alive. To remedy the situation you will need root access on your workstation. There's no problem if you are in a public cluster. If not, then contact the person who maintains the workstation.

To restart your hostmanager program, you need to become "root" and start the program:

```
athena% cd /bin/athena
athena% su                     become superuser
Password: mrroot               enter the password
athena# zhm &                        start new host manager
athena# exit                         exit from the root shell
athena% zctl load              reload subscriptions
athena% cd                     back to your home directory
```

Duplicate Windowgrams

If you find that you are receiving copies of every Zephyr message you get, you most likely have two **zwgc** programs running. To see, use the **ps** command:

```
athena%  ps aux | grep zwgc
```

You will see something like this:

```
costello 26266  2.5  3.9  337  244 p1 S     0:00 zwgc
costello 25855  1.8  5.2  385  325 v0 S     0:38 zwgc
```

To get rid of one of them, use the number in the second column, like so:

```
athena% kill 26266
```

(You can type the **ps aux | grep zwgc** line again to see what you have left.)

To avoid this happening, make sure you are not starting **zwgc** in any of your "dotfiles" like ~/.startup.X. The Athena system files automatically start **zwgc** for you when you log in. Also, if you are experimenting with Zephyr customization, you don't need to start a new **zwgc** every time you want to incorporate your new changes. To incorporate changes to your .zwgc.desc without waiting for the next login, just type:

```
athena% zctl wg_read
```

To incorporate changes to ~/.Xresources without waiting for the next login, you need to type the following line, then kill and restart **zwgc**:

```
athena% xrdb -load ~/.Xresources
```

Other Sources of Information About Zephyr

OLC Answers

The Athena Consultants maintain a set of answers to commonly asked questions about the Athena system. To see the answers relating to Zephyr, see the topic Answers to Common Questions about Zephyr in the Zephyr menus of the On-Line Help facility.

Man Pages

There are Unix Manual pages available on several Zephyr topics. To see any of them, type a command of the following form at the `athena%` prompt:

man *topic-name*

where *topic-name* is one of the following:

```
zaway          zleave          zwgc
zbrowser       zlocate         zwrite
zctl           zmailnotify
zephyr         znol
```

For more technical topics, type:

man 8 *topic-name*

where *topic-name* is one of the following:

```
zephyrd        zinit          zshutdown_notify
zhm            zpopnotify     zstat
```

Discuss Meeting

There is a Discuss meeting called Zephyr_Comments (zcmts for short) for discussions about Zephyr. To add this to your list of meetings, type the following within **discuss**:

```
discuss: am  menelaus:/usr/spool/discuss/zcmts
```

See the information in the On-Line Help facility for more information about using Discuss.

Table of Contents