

**Zeller**

Copyright © 1997 Stephen Williams

---

**COLLABORATORS**

	<i>TITLE :</i> Zeller		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 9, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Zeller</b>	<b>1</b>
1.1	main . . . . .	1
1.2	Zeller 2.0 Introduction . . . . .	1
1.3	Zeller 2.0 Usage . . . . .	2
1.4	Zeller 2.0 Tooltypes . . . . .	3
1.5	Zeller 2.0's ARexx Commands . . . . .	4
1.6	Zeller 2.0 Math . . . . .	5
1.7	Zeller 2.0 Limitations . . . . .	6
1.8	Zeller 2.0 Acknowledgements . . . . .	7
1.9	Zeller 2.0 Program History . . . . .	7
1.10	About MUI . . . . .	9
1.11	Zeller 2.0 Legal Matters . . . . .	9
1.12	Zeller 2.0's Author . . . . .	10

---

# Chapter 1

## Zeller

### 1.1 main

Zeller 2.0

by  
Stephen Williams

Release date: December 14, 1997

Introduction

Usage

Tooltypes

ARexx Commands

About Zeller's Congruence

Limitations

Acknowledgements

Program History

About MUI

Legal Matters

### 1.2 Zeller 2.0 Introduction

What is Zeller?

A perpetual calendar based on Zeller's Congruence. It's basically a program to display any month from any year between AD1753 and AD2100, using the Gregorian calendar. The Gregorian calendar is in use all

---

over the Western world, and was 'invented' by Pope Gregory in AD1585. It is based on the Julian calendar, invented by Julius Caesar.

What isn't Zeller?

Zeller is not an appointment scheduler, personal organizer, or anything of that ilk. It's for reference only.

How to install Zeller

Drag the icon to wherever you want to run Zeller from. If you want to put it in your WBStartup drawer (to display the current calendar whenever you reboot) then please make sure that the icon has the DONOTWAIT tooltype set.

Zeller requires AmigaOS release 2.00 or greater (Exec version 36 or greater). It uses the

MUI

toolkit for its interface, and thus

requires a full MUI installation, with muimaster.library version 11 or greater.

Next section:

Usage

### 1.3 Zeller 2.0 Usage

Previous section:

Introduction

How to use Zeller

Zeller can be run from Workbench or Shell. When the program first starts, you will see two windows; the calendar for the current month in one, and a control panel in the other. Zeller reads the system date; if you have no battery backed clock, I cannot guarantee which month Zeller will display initially.

The most important gadgets in the control panel are the slider and the cycle gadget. You use the cycle gadget to select the month to display. The slider is used to select the year; it's calibrated in steps of one year, from AD1753 to AD2100. Alternatively, you can type the year you want into the string gadget on the right of the slider. Once you've selected the month and year you want, clicking the 'Update' button will actually display the calendar for this month.

Clicking on the 'Print' button dumps the calendar display to the current Preferences printer. You will be given the chance to confirm that you really do want to print; if you have no printer connected then Zeller will appear to hang. The printing I/O is designed to timeout after twenty seconds, so if Zeller hangs when you try to print, leave it for a few seconds and it should wake up again. Zeller

doesn't open the printer.device until you click the 'Print' button for the first time. Once the printer is open, Zeller hangs onto it until you quit.

Clicking 'Reset' resets the calendar to the month originally displayed. The gadget is disabled if the display is currently showing the original month.

Clicking 'Hide Controls' closes the control panel window. This is the same as closing the control window.

Clicking 'Quit' exits Zeller. This is the same as closing the calendar window.

Zeller has a keyboard shortcut to go with each button; the key is indicated by underlining the appropriate letter in the button labels. There is also a TAB selection ring; pressing the TAB key cycles through the gadgets; pressing the RETURN key activates the gadget.

Zeller has a menu strip, the same strip is attached to both windows. The "Project" menu contains the following items:

```
+-----+
| * Controls | This is a boolean item which hides or shows the controls
+-----+
| About...   | This brings up the obligatory information requester
| Quit       | This quits Zeller
+-----+
```

The "Settings" menu contains the following item:

```
+-----+
| MUI...     | Runs the MUI Preferences program to set options for Zeller
+-----+
```

Next section:

Tooltypes

## 1.4 Zeller 2.0 Tooltypes

```
Previous section:
Usage
Zeller's Tooltypes
```

Zeller responds to 4 tooltypes. They are always read, whether the program is started from Shell or Workbench.

AUTOUPDATE

Setting this tooltype will make the calendar update automatically whenever a control is changed. The 'Update' button will not appear on the control panel. This makes the program more intuitive, but it will slow to a crawl on slow Amigas whenever you use the slider. I do not

---

recommend you use this on a machine slower than an A1200 (020/14) with Fast memory.

#### MULTIWARN

Zeller can detect multiple running instances of itself. Set this tooltype if you want Zeller to warn you if it is the second or subsequent running instance (useful if you iconify Zeller and forget about it, then run another copy).

#### NOCONTROLS

Set this to disable the control panel at startup. It can of course be enabled using the menu. You may want to set this option if you put Zeller in your WBStartup drawer; you will then get the current month displayed on every bootup. If you do this, you should set the DONOTWAIT tooltype to inform Workbench to carry on with its startup, even though Zeller is still running.

#### NOMUIREQ

Set this tooltype to use Intuition requesters instead of MUI ones.

Next section:

ARexx

## 1.5 Zeller 2.0's ARexx Commands

Previous section:

Tooltypes

ARexx Commands

MUI

gives a free ARexx port to a program, and Zeller listens on it for some commands. All the usual MUI commands (SHOW, HIDE, QUIT etc) are supported. In addition, you can use the following:

#### HIDECONTROLS

Hides the control window. Equivalent to clicking the Hide button in the control window, clicking the control window's close gadget, or deselecting the Controls menu item.

#### SHOWCONTROLS

Shows the control window. Equivalent to selecting the Controls menu item.

#### TOGGLECONTROLS

Toggles the status of the control window.

---



ABOUT

Displays the About requester.

PRINT

Prints the current month. Equivalent to clicking the Print button in the control window.

There is also one more ARexx command, but I'm not going to tell you what it is or what it does :-). (Don't worry, it's not dangerous).

Next section:

About Zeller's Congruence

## 1.6 Zeller 2.0 Math

Previous section:

ARexx

About Zeller's Congruence

This section describes the math used in the program. Only read it if you are interested!

Any year number can be divided into two parts: the 'Century' part, and the 'Year' part. These are calculated as follows (integer arithmetic, fractional parts of decimals are junked):

$$\begin{aligned} \text{Century} &= (\text{YYYY} - 1) / 100 \\ \text{Year} &= (\text{YYYY} - 1) - (\text{Century} * 100) \end{aligned}$$

where "YYYY" is the original year, in standard four-figure notation.

Using the year 1977 (the year of my birth) as an example, we get the following values out:

$$\begin{aligned} \text{Century} &= (1977 - 1) / 100 \\ &= 1976 / 100 \\ &= 19 && \text{[integer arithmetic, remember]} \\ \text{Year} &= (1977 - 1) - (19 * 100) \\ &= 1976 - 1900 \\ &= 76 \end{aligned}$$

Therefore, for Year = 1977, Century = 19 and Year = 76. Note that these are not what we would ordinarily think of as being the century and year; we'd normally call the year "77" and the century "20".

Zeller's Congruence makes use of these values to calculate the day on which January 1 of any particular year falls. The exact formula is as follows (again, this is integer arithmetic; any remainder from the divisions is ignored):

$$\text{day} = (799 + y + (y / 4) + (c / 4) - (2 * c)) \bmod 7$$

In the formula, *c* represents the Century number (19 in our example), and *y* represents the Year number (76 in our example). The result will be an integer between zero and six. We take zero to mean Sunday, one to mean Monday, etc. up to six to mean Saturday.

Our example comes out as follows:

$$\begin{aligned} \text{day} &= (799 + 76 + (76 / 4) + (19 / 4) - (2 * 19)) \bmod 7 \\ &= (799 + 76 + 19 + 4 - 38) \bmod 7 \\ &= 860 \bmod 7 \\ &= 6 \end{aligned}$$

So we can deduce that January 1, 1977 was a Saturday. From this, it is easy to display the calendar for January.

The other months are a little more tricky. Take February. Again, we want to find out which day February 1 fell on. The procedure is as follows:

- Take the number of the day January 1 falls on (6 in this case);
- Add to this the number of days that elapse between January 1 and February 1 (31 days);
- Take this number mod 7 ( $37 \bmod 7 = 2$ );

This is the number of the day February 1 falls on (day 2 = Tuesday).

The same method can be used for any month, using a different number of elapsed days. There is a complication: leap years. A leap year occurs whenever the year number is divisible by four (eg 1704, 1980), except if the year is a century year. In this case, it is a leap year if divisible by 400 (so 1600 was a leap year, but 1700 was not). For leap years, the number of days that have elapsed between January 1 and any month past February will be one more than non-leap years.

Next section:

Limitations

## 1.7 Zeller 2.0 Limitations

Previous section:  
About Zeller's Congruence  
Bugs and Limitations

Zeller's Congruence operates correctly from the year AD1585-ish; this is the year in which Pope Gregory 'invented' the current calendar. Therefore, making the program's range begin at AD1753 could be seen to be a limitation. Zeller begins in AD1753 because the Gregorian calendar was adopted in England in September of AD1752, so AD1753 was the first full year that the calendar was in use. The next release of Zeller will address this.

The ARexx interface is less than useful; it should be possible to

---

issue commands to print a whole batch of months, eg. for a whole year. Will be addressed.

One beta-tester reported that the NOMUIREQ tooltype didn't work on his system. I'm mystified; it works fine on mine, and I can find no problems with the code. Zeller doesn't monitor the tooltypes array for changes; it is only read at program startup. If you want to change from MUI to Intuition requesters or vice-versa, you must quit Zeller and restart it to notice the change.

This is my first MUI program, so there are bound to be oodles of problems.

Next section:

Acknowledgements

## 1.8 Zeller 2.0 Acknowledgements

Previous section:

Limitations

Acknowledgements

I would like to thank the following intrepid beta-testers for agreeing to test this program for me.

Indy2 from #amiga (sorry I don't know your real name :-)  
Dave Hardenbrook  
Chris Underwood  
Gernod Schomberg  
Glen Watts

Without them, Zeller would contain more bugs than a Volkswagen dealership.

Zeller was inspired by an article by Mark Smiddy in Amiga Shopper magazine, which detailed a perpetual calendar for AmigaDOS script language. My program (written in HS Pascal) uses the same math as his program. This was reproduced without permission; but hey, how many ways are there to implement Zeller's Congruence? :-)

Next section:

Program History

## 1.9 Zeller 2.0 Program History

Previous section:

Acknowledgements

Program History

---

- 
- Zeller 0.1 (August 30, 1996; beta release)
    - First release.
  
  - Zeller 0.2 (October 6, 1996; beta release)
    - Repaired small bug in code to write initial month (only happened under AmigaOS release 2.x). Thanks to Dave Hardenbrook for reporting this bug.
  
  - Zeller 0.3 (December 11, 1996; beta release)
    - Rewrote code which reads system date; now smaller and more efficient.
    - Added 'Reset' button.
    - Optimized gadget rendering code.
    - 'Shrink' button added to title bar.
  
  - Zeller 1.0 (March 17, 1997; first public release)
    - Public release of v0.3.
    - Zeller now uses the screen font, if it is a fixedwidth, eight-point font.
    - Added some extra checks to initialization code.
  
  - Zeller 2.0B1 (July 24, 1997; beta release)
    - Almost total rewrite.
    - GUI toolkit changed from GadTools to MUI. Program structure made object-oriented.
    - Year range changed to AD1753 -> AD2100.
    - User interface split into two windows (calendar and controls).
    - Appearance of printouts improved.
    - Tooltypes added.
  
  - Zeller 2.0B2 (September 9, 1997; beta release)
    - Now highlights today's date. This should have been in there from the beginning, but it's one of those things I never got around to adding until now.
    - Printing I/O made more efficient.
    - Printing I/O now times out after twenty seconds; no more hung Zellers if you try to print something and your printer doesn't work/isn't connected.
    - Lots of small inefficiencies corrected.
    - Fixed a bug in the `_documentation_` that was crashing `amigaguide.datatype` with alert `0x81000005` or `0x8100000C`. Sheesh. One misplaced quote mark and the memory list falls apart :^)
  
  - Zeller 2.0B3 (October 5, 1997; beta release)
    - Now calculates system date using `utility.library` functions instead of my own (bugged?) code.
    - Added ARexx commands.
    - Added keyboard shortcuts and TAB gadget selection ring. Hopefully, Zeller now supports everything in the MUI style guide :-)
  
  - Zeller 2.0 (December 14, 1997; public release)
    - Public release of 2.0B3. Some small changes:
      - Printer output prettied up a little;
      - "Settings" menu added;
      - Zeller now tries to use it's own icon when iconified using MUI;
      - Bug fixed in ARexx setup;
      - "Hide" button renamed to "Hide Controls", as this is a more accurate
-

description of this button's function.

Next section:

About MUI

## 1.10 About MUI

Previous section:

Program History

About MUI

MUI coders are supposed to include a little plug, so here goes...

MUI (Magic User Interface) is a font-adaptive, user-configurable toolkit for creating and maintaining GUIs. The behaviour of a MUI user interface is under complete control of the user, not the programmer.

MUI under continual development by Stefan Stuntz, and is published by SASG. It's a shareware product, available from the SASG website <http://www.sasg.com/>

You don't have to register to use it, but I recommend you do; it gets a lot better (more configurable) if you register.

MUI is copyright © 1992-1997 by Stefan Stuntz.

Next section:

Legal Matters

## 1.11 Zeller 2.0 Legal Matters

Previous section:

About MUI

Legal Matters

Zeller was inspired by an article by Mark Smiddy in Amiga Shopper magazine, which detailed a perpetual calendar for AmigaDOS script language. My program (written in HS Pascal) is partly based on his code (see

Acknowledgements

).

Zeller is © 1996, 1997 by Stephen Williams.

Zeller is not in the public domain. The author, Stephen Williams, holds the copyright. Zeller may not be altered in any way. No profit may be made from Zeller.

---

This version of Zeller is a public release. It may be distributed by FTP, on bulletin boards, by PD libraries, or on magazine cover (disks|CDs). The author is reasonably satisfied that it is bug-free, stable, and won't destroy your hard drive on a whim.

The author may not be held responsible for damage, loss, injury or squirrel maiming caused as a direct result of using Zeller.

Back to top

## 1.12 Zeller 2.0's Author

How to contact the author

Send snail-mail to:

Stephen Williams,  
9, Helmsley Way,  
Spalding,  
Lincolnshire,  
PE12 6BG,  
England.

During University term time, contact me by:

email: S.Williams@warwick.ac.uk  
IRC: Steffan on ANet #amiga  
(This account expires in June 1998)

Other software by Stephen Williams

Cookies v4.5 (Aminet util/misc/Cookies.lha)

A fortune cookie engine, with a ReqTools or Shell interface. It's easy to customize with new fortunes, and has PowerPacker and XPK support.

UnSquish v2.0 (Aminet util/pack/UnSquish.lha)

A universal decruncher. It uses xfdmaster.library, and so supports most common (and not so common) compressors.

Condition: Green! (Aminet mods/med/condgreen.lha)

A 4-channel MED module.

Switch v1.2 (Aminet util/misc/Switch.lha)

An Intuition based metric conversion utility, which can be used to convert between most common metric and British measures.

---

File v1.0 (Aminet util/misc/File.lha)

An Amiga version of the Unix "file" command, which uses FileID.library to identify most file types. It also offers a drag and drop interface when started from Workbench.

---