GNUCash Group 2

• • •

Bella, Eric, Liandra, Raja and Ryan

Project objective:
Apply software maintenance practices learned in class to GNUCash.

Overview

Working as individuals, each member of the team chose 1 bug from bugzilla to reseach and work on. Team members are expected to guide their own development for the fix, as well as provide support for other team members in problem solving and issue resolution.

Each team member was encouraged to go a bit out of their comfort zone, but to have a backup issue in case of overreaching.

Agile-Like Development Process

Bi-Weekly Meetings

In order to emulate the Scrum methodology in a way that works for our team, we are having 15-minute meetings after class, rather than a daily stand-up.

Weekly Iterations

With the small development window of only a few months, iterations every 2 weeks will not work. Instead we are focused on having weekly work products.

Flexible Schedule

Being students, our priorities can shift on a daily basis. As a result, we are keeping a flexible schedule and not fretting about due dates. Quality work is stressed over arbitrary team deadlines.

Collaboration Tools

Github

Git, through Github, is being utilized to allow for version control and file sharing for peer reviews. We chose to use a personal github page before branching from the GNUCash to allow for a judgement free work environment

Slack

Since we are unable to meet everyday, having a clear and near-constant line of communication is useful. Slack allows use to create that discussion space.

Google Docs

For documentation and creating presentations, we are using Google docs. This allows for everyone to have access to the documents we are working on.

635708 - Report Style Differences - Overview

Reported Issue

A user reported the lack of a visual difference between the invoice style sheets Easy, Footer, fancy and Technicolor. Further investigation has shown that style templates create different visual reports in Tax invoices or Fancy invoices. Even more confusingly, the style sheets are based on similarly, but not identically, named templates. While this is not necessarily a high severity issue, it is confusing to new users and has stuck around since 2010.

Affected Areas:
- Creating Transaction Reports
- Creating Printable Invoices

Style Sheet to Template Chart

Template Name	Sheet Name
Easy	Easy
Fancy	Technicolor
Footer	Footer
Plain	Default

635708 - Report Style Differences - RCA

Possible Root Cause

The history of the templates shows that Plain and Fancy were added in 2000. Easy was later added in 2004, as part of an "Easy Invoice" patch, but is indistinguishable from the Fancy style sheet. Footer was then added in 2010, based on the Easy template but with an added footer section. Despite Footer being functionally identical and backwards compatible with Easy and Fancy, both of them have stuck around as development artifacts.

The root of this bug is not a typo nor an issue with the logic in the code, instead it is a lack of oversight in the implementation of unneeded style sheets.

635708 - Report Style Differences - Road Map

Development Steps

- Remove Fancy and Easy Templates
- Create hidden aliases for Fancy and Easy that point to Footer, allowing for backwards compatibility
- Rename Footer to something more generic

Integration/Testing Steps

- Ensure Fancy and Easy do not show up as options
- Ensure Fancy and Easy reports still work with the created alias
- Integration should not require much work as the sheets are not linked/called outside of the reporting sections

635708 - Report Style Differences - Current Status

Creating Hidden Aliases

Currently investigating how GNUCash reads and opens old reports. Further investigation is also going into what other files touch on the style sheet templates. The goal is to anticipate future issues with the make-files and headers.

635708 - Report Style Differences - Still to Do

Rename Footer

New Name

- Instead of using suggest Fancy name, using the label Deluxe. Will allow for clarification and no issues with alias pointing.

Clarify Documentation

- Ensuring that the naming issues do not cause future issues, the documentation for the reports features needs to be updated with the new names as well.

Testing/Integration

Testing

- Creating test cases will take some time, but passable tests should be ready around mid-November.

Integration

- Full GNUCash integration will be difficult to reach by the end of November, but not out of the realm of possibility

569749 - Company Name Fields Not Populated - Overview

Reported Issue

The company name from the book options (File -> Properties) should automatically populate the company fields all the financial reports. ie Balance sheet, income statement, cashflow statement and changes in equity statement.

Currently it has to be manually entered every time a report is run.

569749 - Company Name Fields Not Populated - Root Cause

Possible Root Cause

Even if the company name is prefilled in the file properties, the Company Name is not prefilled in the Report. There is no actual fault with the output but the correct output is not being displayed due to a connectivity problem. This is now having users manually fill in the company name in various fields for the balance sheet, income statement, cash flow statement, and changes in equity statement.

569749 - Company Name Fields Not Populated - Road Map

Development Steps

- Ensure inheritance of classes is correct and variables/methods are public
- Create variable/use existing variable from file properties input and connect to empty fields requiring company name.

Testing/Integration Steps

- Unit testing and integration testing
- Static testing will confirm from the user perspective that classes connect properly
  - Ensure that all fields in all sheets populate automatically once field properties is completed.

569749 - Company Name Fields Not Populated - Status

Testing Connectivity

- Currently testing the connectivity of classes and understanding methods and how they are called in derived classes.

569749 - Company Name Fields Not Populated - What's Left

Manually populating fields with Company name

- Connecting fields
- Reviewing documentation
- Testing

692831 - Bad Rendering on Retina Displays - Overview

Issue to Report

- Apple SDK is extensive, lot to sort through, but I am learning a lot
- The Virtualbox has its own graphic engines on MacOS, which needs to be accounted for

- Both images (.png) and text are pixelated, and the SDK processes them both differently.

692831 - Bad Rendering on Retina Displays -RCA

Possible Root Cause

Bug 692831 was first reported on January 9th, 2013 (right after Apple unveiled the first Retina Display). The issue is not a error in the present build, but an oversight in software maintenance, as the developers did not prepare for or update the build to accommodate the new technology.

692831 - Bad Rendering on Retina Displays - Road Map

Development Steps

- Identify Apple Graphic SDK
- See if current font schemes and graphics can be modified to conform to the SDK
- If not, become an electronic cryptologist

Integration/Testing Steps

- Ensure the running program is not pixelated for the end user
- Ensure modifications do not affect the user experience

692831 - Bad Rendering on Retina Displays - Current Status

35% Progress

Currently investigating the difference between the virtualbox graphics engine and the native MacOS High Sierra Build. So far, I have graphics displaying properly within the virtual box, but then it is lost when running on Native MacOS. I believe this is due to the resolution change in between both Operating Systems, as on the virtual machine I have HiDPI set to unscaled, which is another Apple Marketing Technique.

This weekend, and next week, I want to use the SDK to modify the HiDPI settings for the build, thus hopefully fixing the issue.

692831 - Bad Rendering on Retina Displays - Still to Do

Learn about HiDPI Manipulation

VirtualBox

- Figure out why HiDPI handles the modifications differently

MacOS High Sierra

- Modify HiDPI drivers and/or code to receive the changes made to the build running on Ubuntu

Testing/Integration

Testing

- Test Builds on both VirtualBox and MacOD High Sierra

Integration

- Propose changes to the production build, and go through the acceptance process.

782423 - Setting Preferences is outdated - Overview

Issue to Report

A user noticed that section 10.2 of the user documentation, titled "Settings Preferences" is outdated. The user reported this issue and linked a direct comparison between the "Preferences" found in the actual software and the interface shown in the user documentation. This may not seem like the biggest issue, but an accurate and up to date user documentation is the backbone to any robust program, and the lack off may lead to user confusion and issues with future maintenance

Affected Areas:

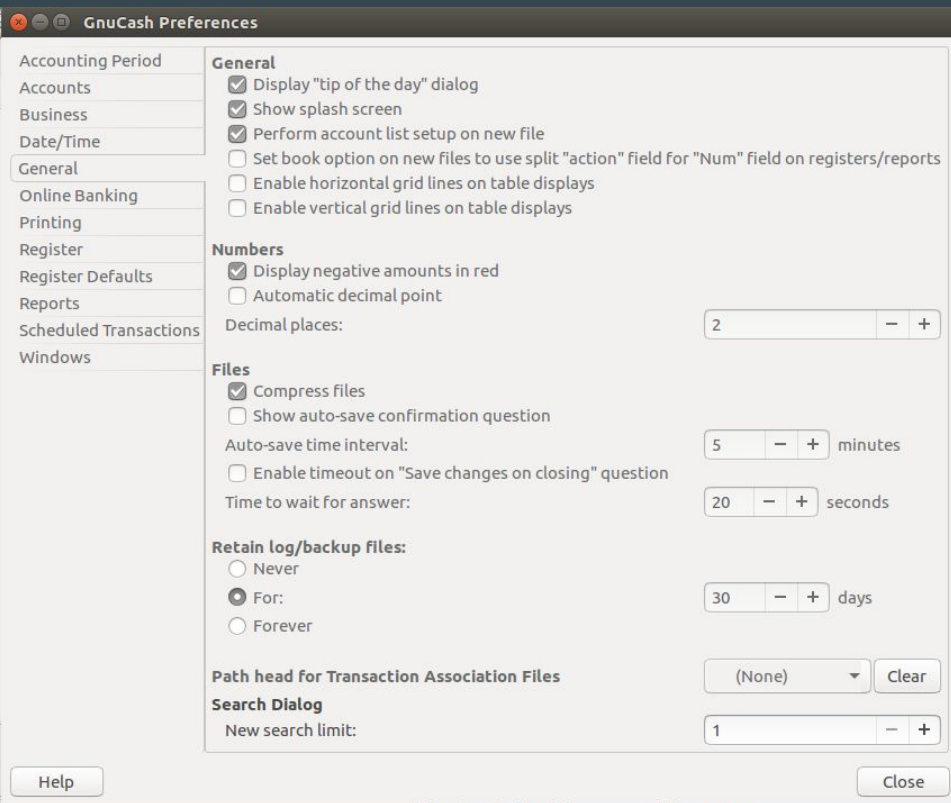- User Documentation section 10.2

Timeline:

- Reported: 2017-05-10
- Modified: 2017-07-30

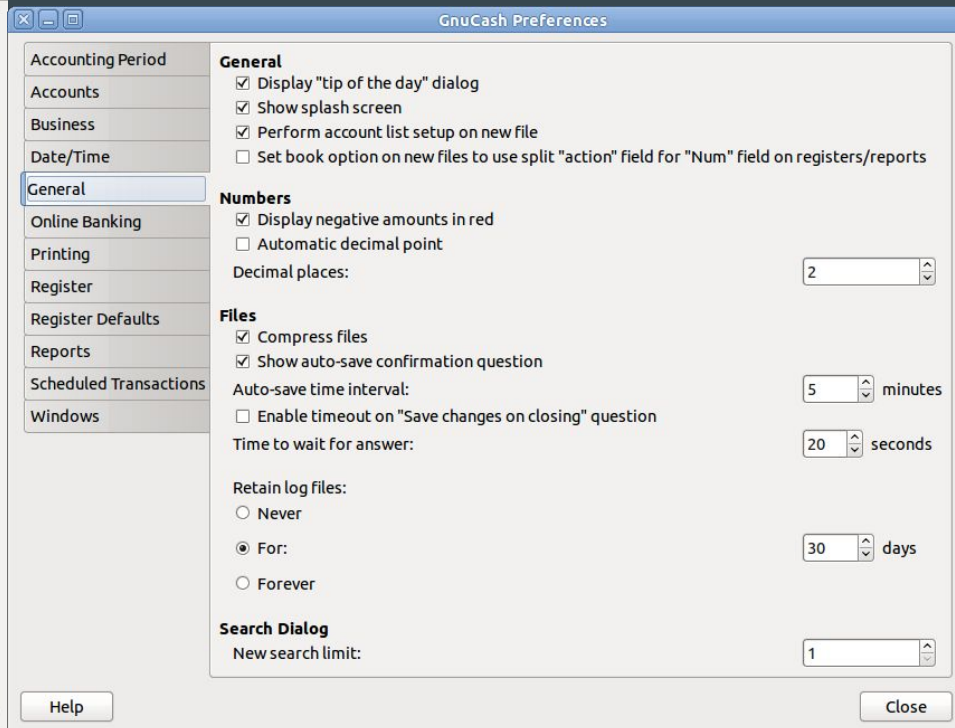# 782423- Setting Preferences is outdated - RCA

Root Cause

The root cause may stem from a lack of resources. As simple as the issue might be, updating documentation is time consuming and may not be given the same priority as more code-breaking issues. Since there is no automatic tool to update documentation, the developers must have been updating it manually as each change is made. A program such as GnuCash that gets maintained regularly requires the documentation to be maintained alongside of any changes, and with issues stacking up, this part of the process may have been forgotten

782423- Setting Preferences is outdated - RCA

GNUCash

Help Example

782423- Setting Preferences is outdated - Road Map

Development Steps

- Clone additional documentation repository
- Remove old images from the documentation
- Remove outdated text
- Take new screenshots that have the same dimensions as the old ones
- Add new descriptions and upload new images

Integration/Testing Steps

- Launch the new webpage and ensure that the formatting and text are error free
- Compare the new version side by side with the old version to see if everything is correctly formatted
- Team review to proof read the new html document
- Since the issue is html based, with very little actual code involved, other forms of testing may not be feasible

782423- Setting Preferences is outdated - Current Status

HTML, Building and editing

Currently understanding html syntax, and the build process for a web page. Also comparing searching for a linux compatible IDE that will fit our needs best, as i have only played around with html directly from a browser so far.

782423- Setting Preferences is outdated - Still to Do

Updating Documentation

Formatting

- A bulk of the difference between the old and new lies in the new features. Figuring out a format for adding new content that fits with the old

Coding

- All Html code must follow the same standards that the rest of the document follows. The coding should be done before the first week of november

Testing/Integration

Testing

- Since most of the testing is reviewing and proofreading, this should be completed within 2 days of code completion

Integration

- We will be pushing for integration before mid November

782423- Setting Preferences is outdated - Whats Next

Be a team player

Early completion of the bug will be followed by aiding other team members with their work. The level of difficulty with each bug varies, and fixing individual bugs is no more important than accomplishing the goals of the team

331739 - Scaling Report Graphs

Issue:

Users reported scaling issues when creating charts for expenses, budgets, income etc. Once a chart has been made it cannot be resized, and size does not change when the window is resized. Scaling should be automatic

Root:

No root cause except for that the option to rescale does not exist. Report options allows the user to choose the dimensions before the chart is created, but after

Development Steps:

Create hierarchy account and subaccounts, create sample expenses/budget

Current Status:

Not a bug, more of a feature to make the user be able to scale easier, instead of having to enter dimensions

What's Next:

Next Bug