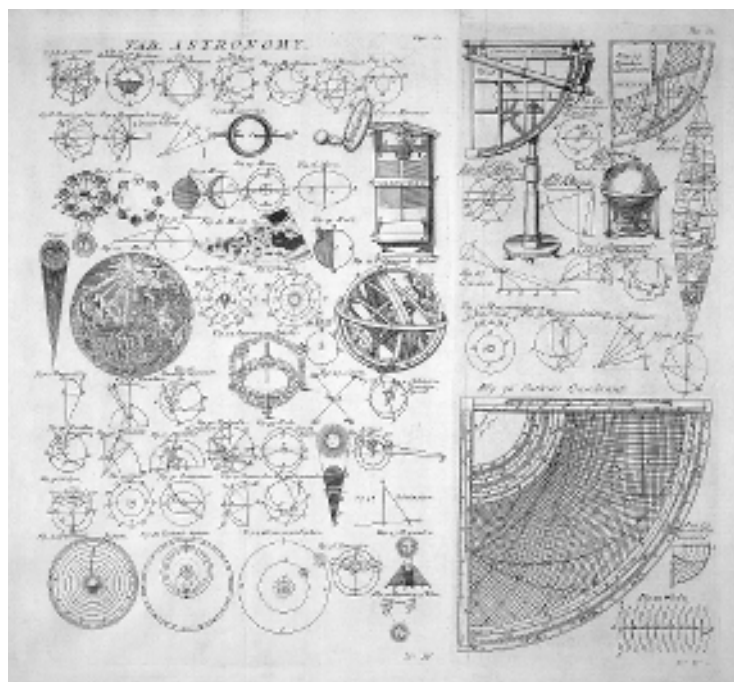


# COS Astrolab

## v0.9866\_20140425

**A guide to development of the App.**



**Copyright**

Copyright in this document is the property of C. J. Gaudion.

© **Copyright 2014**

## Contents

<b>1. Abstract.....</b>	<b>6</b>
<b>2. Introduction .....</b>	<b>7</b>
<b>3. The Story so far .....</b>	<b>8</b>
<b>4. Summary of the Development Process.....</b>	<b>10</b>
4.1. Development Cycle .....	10
4.2. Current Position for COS Astrolab development .....	13
<b>5. Detailed Development Process .....</b>	<b>15</b>
5.1. Initial Ideas .....	15
5.2. Basic requirements of this project .....	15
5.3. Why re-invent the wheel? .....	15
5.4. How difficult will this be? .....	16
5.5. Why develop in Open source + Cross Platform.....	16
5.6. Initial Ideas Summary .....	17
<b>6. Current Astrolab Version is 0.9864_20140325.....</b>	<b>18</b>
<b>7. Installation .....</b>	<b>18</b>
<b>8. Starting the Application.....</b>	<b>18</b>
<b>9. Application Contents .....</b>	<b>19</b>
<b>10. Develop .....</b>	<b>20</b>
<b>11. Astrolab Application Build for Distribution Process .....</b>	<b>22</b>
<b>12. Phase – Planning.....</b>	<b>25</b>
12.1. Stage – Scope .....	25
<b>13. Phase – Requirements.....</b>	<b>26</b>
13.1. Stage – Specification for Platforms used in Development.....	26
13.2. Stage – Detailed Specification – Supporting Web Site.....	27
<b>Phase – Analysis and Design .....</b>	<b>28</b>
<b>14. Phase – Implementation .....</b>	<b>28</b>
<b>15. Phase – Deployment .....</b>	<b>28</b>
<b>16. Phase – Testing .....</b>	<b>28</b>
<b>17. Phase – Evaluation.....</b>	<b>28</b>
<b>18. Appendix A – Initial Ideas - ‘To Do’ List for COS Astrolab.....</b>	<b>29</b>
<b>19. Appendix B – Initial Ideas - ‘To Do’ List for COS Website.....</b>	<b>30</b>





## **1. Abstract**

The COS Astrolab web application and its supporting website is intended as far as possible to produce a one-stop resource for astronomy, including data, calculations, and general information. The web site will also contain information on selected good open source software tools, or free software tools. The project will also document the development and support of the COS Astrolab multi function application that could be of general use to astronomers at all levels, using Javascript, HTML and any easy to use, free development tools, that may be found available. The aim is to use standard data and functions that can be re-used throughout the application as it grows. The documentation will always attempt to describe the simplest way to achieve this to for a wide audience.

## 2. Introduction

This project started out to combine years of experience supporting and developing computer applications with the study of Astronomy and Astrophysics, in an effort to document the development and support of a multi function application that would be of general use to astronomers at all levels. It is worth mentioning right now that when we refer to 'Astronomy' throughout this document we are using the term to cover Astronomy, Cosmology and Astrophysics.

Astronomy, Astrophysics and Cosmology have differing definitions. Astronomy can be described as the observational subject, Astrophysics attempts to explain the observations and Cosmology is the study of the Universe as a whole. As there is much overlap between in the subjects in books and courses, but as stated above, we will usually describe everything under 'Astronomy' for the purposes of this project. We have however included brief, but more detailed definitions below.

**Astronomy** is the science of celestial objects and phenomena that originate outside the Earth's atmosphere, such as stars, planets, comets, auroras, galaxies, and the cosmic background radiation. It is concerned with the formation and development of the universe, the evolution and physical and chemical properties of celestial objects and the calculation of their motions. Astronomical observations are not only relevant for astronomy as such, but provide essential information for the verification of fundamental theories in physics, such as general relativity theory. Complementary to observational astronomy, theoretical astrophysics seeks to explain astronomical phenomena.

**Astrophysics** is the branch of astronomy that deals with the physics of the universe, including the physical properties (luminosity, density, temperature and chemical composition) of astronomical objects such as stars, galaxies, and the interstellar medium, as well as their interactions. Because it is a very broad subject, astrophysicists typically apply many disciplines of physics including, but not limited to, mechanics, electromagnetism, statistical mechanics, thermodynamics, quantum mechanics, relativity, nuclear and particle physics, and atomic and molecular physics. In practice, modern astronomical research involves a substantial amount of physics. The name of a university's department ("astrophysics" or "astronomy") often has to do more with the department's history than with the contents of the programs.

**Cosmology** - the study of cosmology is theoretical astrophysics at the largest scales, i.e. the universe as a whole.

### Putting it all in an 'App'

I know from my own experience starting programming in FORTRAN, while studying chemistry, I would probably write code to do the task I want, in roughly the user interface format I want. As long as it works then I am not going to worry too much about any particular programming trend at the time. A lot of software in science has probably started like that, so there are lots of bits and pieces out there, but there does not appear to be an equivalent of the 'Excel / Word' suite of programs that covers most of the functions in an office.

However, the range of activity in Astronomy and associated fields is pretty wide – number crunching, data reduction, image processing, factual information and more. There are programs like Stellarium for observing and data on celestial objects, FITSLiberator for image analysis, MATLAB for calculations, but the effort and programming knowledge to combine these sort of functions into a single suite of programs would be huge, never mind any copyright issues.

So what alternatives are there? Well the one we looking at is to document the most useful existing software and other resources for astronomy in a single web site. At the same time

we can try and develop one application under this supporting web site, to try and cover some useful functions that are practical to program without a lot of effort.

One major requirement in this project is to minimize cost, we do not want to have to be buying programming software or spending money on professional software like Photoshop which can cost a small fortune. This project is also aimed at students, amateur astronomers, or anybody with an interest in astronomy, so keep the cost near to zero. How do we do that? Well we use Open Source software community and free tools like HTML and Javascript that are built into your computers web browser software.

The rest of this document will try and provide a useful guide on how to run this sort of project and how the development of the application is done and how the application works.

### **3. The Story so far**

The project has so far been developing in two parts as detailed below. This all started off in a 'Windows XP' environment, then switched to iMAC about 3 years ago, so we will be looking at continued progress from an OS X view. However as its all HTML and Javascript, then its basically text files so you can run and amend it under other operating systems.

The whole project started with two parts:

1) The Conquest of Space web site attempts to detail open source software and document the best resources for the study of Astronomy generally, as well as those that compliment the Astrolab application and its development.

The name 'Conquest of Space' was chosen to cover the exploration of the cosmos as a whole and comes from a 1955 science fiction film produced by George Pal which depicts a voyage to Mars. The science and technology were intended to be as realistic as possible. The poster tagline was "See how it will happen in your lifetime!" The film includes an orbiting space station and a mission to mars – two subjects currently being covered in our lifetime!

2) the standalone Astrolab application which runs using HTML, Javascript and some JAVA.

At the same time we are attempting to document how both the website and application all works, to aid those who are interested in contributing, and to show how a useful working application for astronomy could be developed by anyone with little previous knowledge of software development.

This document contains a description of the whole Conquest of Space resource project as well as requirements and 'To Do Lists' for the website and the Astrolab application development, this cuts down on the number of different documents that can often be generated in a large formal project, and we have no wish to bore readers if it can be avoided.

The development process chosen for this project is these days referred to as 'Agile'

In February 2001, 17 software developers met at the Snowbird, Utah, resort, to discuss lightweight development methods. They published the Manifesto for Agile Software Development to define the approach now known as agile software development. Some of the manifesto's authors formed the Agile Alliance, a non-profit organization that promotes software development according to the manifesto's principles.



The Agile Manifesto reads, in its entirety, as follows:

***“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:***

***Individuals and interactions over processes and tools***

***Working software over comprehensive documentation***

***Customer collaboration over contract negotiation***

***Responding to change over following a plan***

***That is, while there is value in the items on the right (of the list), we value the items on the left more.”***

Basically working closely with people who will be using the software, producing working software and responding to changes found to be needed as you go along, is more important than spending ages producing detailed documents on what you are going to do (and in the end the software still does not do what you wanted...). We would however add that in the end some documentation is required - a great user guide!

Agile would seem to suit the amateur developer who learns what he needs to know to do what he wants and will learn by following examples, books and writing code to experiment with – and changing it if needed when it does not work as expected, rather than producing document after document detailing how he is going to do it and if it is like many professional projects, no useful user guide at the end of it – and that’s one of the things the users really wanted.

We have chosen to follow this development route for Astrolab and our methods have changed since the project started, generally aiming to simplify and shorten all the processes as much as possible.

Our development is iterative and we will include a diagram later in this guide to show the ‘flow’. We use ‘to do lists’ and document changes in the code itself and in this user guide.

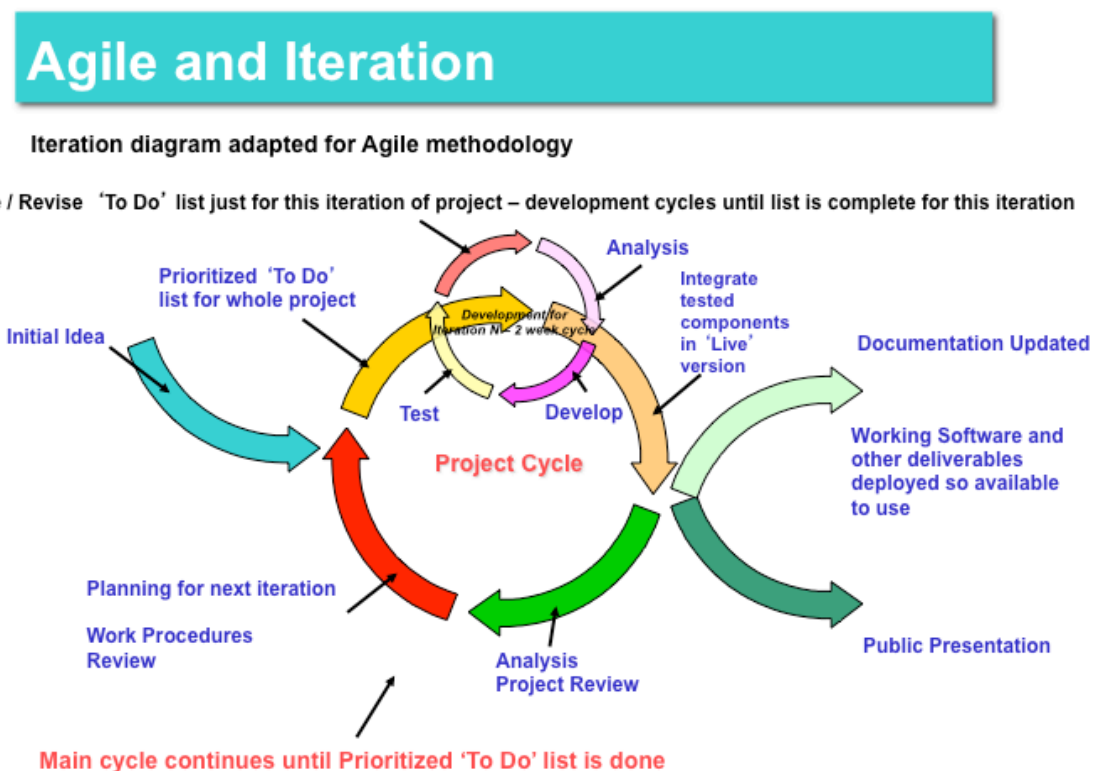
## 4. Summary of the Development Process

### 4.1. Development Cycle

The figure below is adapted from a previous Iterative development diagram; mostly we have just changed the labelling. We are basically going around in cycles, adding items to the project, ending a cycle with something that works in a short period (1 to 2 weeks) and then starting again with another cycle.

**Table 1** later in this guide shows the process below in tabular form, which is included on the supporting website to show the current state of any development cycle.

Figure 1.



#### Initial Idea / Prioritized 'To Do' list for whole project

Basically you come up with an Initial Idea of what you want to achieve and start a 'To Do' list for what and how you want the whole project (be it web site or software application) to do when completed and in what sort of priority the items in the list can be done.

#### Produce / Revise 'To Do' list for iteration

You are not going to write the whole application in one go, so then you produce a list of what can be done say in the next two weeks that will allow further items in the list to be done later (some may require more thought or are harder to code). Now you have an 'Iteration To Do' list for this iteration or 'cycle' within the whole project – give the list a number, would sensibly be the version number shown in final release notes, if this is an application. This way you will get at least something that you can start to use at the end of iteration.

You then work on how you are going to code the first item in the list, write the code, test it and then do the next item in the list and so on until the list is complete.

### **Integrate tested components in 'Live' version**

You then produce a version / package for anybody to test and run. If this is the first iteration there will be nothing else to merge it with. In the case of Astrolab if we add or amend a html page or data source, then we amend the Astrolab manuals / Readme.txt and upload any changes to our live web server under alchymy.com.

### **Documentation updated / Software deployed / Public presentation**

First and most important – any changes are detailed in Readme.txt files that go with the app and any user guides. With agile there are no stacks of documentation just to get it done, the emphasis is on working software – and telling the user how to get best use out of it. So many 'big time' applications lack a decent user guide. Many commercial apps will have a charge for training rather than just produce proper user guides in the first place.

In addition .zip and .dmg files are made available on sf.net and launchpad.net as either current development 'save files', or official releases at a specific version number.

A notification on Twitter etc lets interested users know this has happened.

### **Analysis / Project Review**

Then you can review how the project has progressed and what or how things could have been done better.

### **Planning for next iteration / Work Procedures Review**

Go through your original main to do list and tick items off, maybe new items have been thought of to add, or a different approach that would make the project easier to develop.

The Prioritized 'To Do' list for whole project gets updated and then you pick more items that can be done in say the next two weeks and produce the next iteration to do list, with an incremented version number.

### **Summary**

And so the two overlapping cycles continue until the whole project to do list is complete and so should the application be. Of course in the real world you may come across items that cannot be made to work or new items you want, so the original main list for the project is altered as more 'Initial Ideas' come up and so are later iteration lists. Many open software projects will continue to change and improve as long as there are supporters to do the work.

Appendix A and B in this guide show 'Initial Idea' to do lists for the Astrolab app and the COS website. These lists have probably evolved since the project started. Changes now may just be a list on a post it note or notes on a screen print, but the principle is the same. The end result should be the application and the user guide contains the final details of any changes.

### **'To Do list' for each stage of the cycle**

At present we are just detailing the last stages that get the software out there.

### **Documentation updated**

Details of changes added to user guide and Readme.txt files

### **Working software and other deliverables deployed to sf.net and launchpad.net**

Changes to Astrolab – bug fixes or new features added to 'code'

Changes tested and added to on-line Live version

Development save files in .zip and .dmg. created via script

.zip and .dmg and Readme.txt files uploaded to sf.net

If this is an 'official release' then .zip and .dmg files renamed as

AstrolabV0\_nnn.dmg and AstrolabV0\_nnn.zip

Files uploaded to sf.net and launchpad.net project

### **Public presentation of what has been done**

Notification of new versions on sf.net and launchpad placed on

Twitter via direct tweet or email to twitter >> updates facebook automatically

Update sf.net project news

Update launchpad.net project news

Update Google+ news

## 4.2. Current Position for COS Astrolab development

The table below shows our 'Agile Current Development Status table' that appears on the web site, based on our agile flow diagram, which in turn is based on a previous diagram for 'Iterative software development life cycle'. Our agile process is pretty similar, but has shorter cycles and 'To Do' lists rather than many of the documents encountered in a formal software development environment, in fact we will not even bother going into those documents and methods. In fact the table below is almost a 'To Do' list itself.

These methods and table format can be used to track progress on updates to the supporting web site and individual parts of the whole project, e.g. a Java program, if required.

**Table 1** – COS Astrolab Agile Current Development Status.

	COS ASTROLAB – Version in Progress is:	Version No: 0.9899	
	Stage	Notes	Status
1)	Initial Project Idea / Update existing project	Research, prototyping, 'To Do' list	
2)	'Big To Do' list for whole project	Add to 'To Do' list form 1)	
3)	Revised 'Little To Do List' just for this iteration	'Can do' items this time from 2)	
4)	Analysis	How to code	
5)	Develop	Code	
6)	Test	Test code	<<<<
7)	Integrate	Put into online version	
8)	Deploy	Sf.net	
9)	Document	User Guide	
10)	Present	Twitter / Facebook / sf.net	
11)	Project Review	What next, how did it go?	
12)	Review Work Procedures / 'Big To Do List'	Can anything be done differently, made easier?	
	Return to 2 for next iteration	And Version n+1	

In addition we are using some of the colour coding above to highlight the progress at a particular stage in the table.

This is also being used to highlight the title bars of some of the HTML pages within COS Astrolab, to reflect progress with changes. This gives a useful indication of the state of the development of individual parts of the application, e.g. it works but it's not been tested fully, or still being coded but it mostly works.



## 5. Detailed Development Process

### 5.1. Initial Ideas

The initial ideas for the project require some preliminary assessment and a certain amount of research to basically find out how feasible the idea is, there is no point in jotting down load of ideas for the application to do everything you ever wanted and then get to the programming stage to then find either it just cannot be done because you don't have the knowledge or the development tools are not available or it will be too costly.

We already know we want the application to be web based, open source and run on multiple platforms. The fact that the web site and application exist mean that some development has already started – this brings us to the subject of Prototyping.

*Prototyping is the process of quickly attempting to put together a working model (the prototype) in order to test various aspects of a concept or design, illustrate ideas or features and thus get some idea of how easy the full development will be.*

*Prototyping is often treated as an integral part of the system design process, where it is believed to reduce project risk and cost. Often one or more prototypes are made in a process of iterative and incremental development where each prototype is influenced by the performance of previous designs, in this way problems, or deficiencies in design, can be corrected.*

*When the prototype is sufficiently refined and meets the required functionality and robustness, and the development tools available look like they will be able to cope with the project, then further development can start.*

In this way we are already going around the 'agile cycle' and getting some working software even as we find out if the project is feasible, and at the same time we can be adding the ideas that look feasible to stage – 2) The 'To do' list for the whole project.

The sections following describe some of the investigation done as part of the 'Initial Ideas' and producing the first 'To Do' list for whole project.

### 5.2. Basic requirements of this project

As part of the preliminary assessment we need to expand the idea creation and list the basic requirements of the project i.e. what do we want to achieve, here done as bullet points:

- A web site supporting study of Astronomy and Astrophysics and summarising the best web sites that also support this study.
- A web based application to store basic data on astronomical objects and describe and calculate common equations used in Astronomy and Astrophysics.
- Application to be open source so others can contribute and not be restricted to running on one platform (here platform can be hardware **or** operating system **or** both)
- Web site to also detail and support the hardware and software used in the project and application development
- Document the whole project as far as possible to assist those interested in the subjects covered (including written documentation and a blog).
- Minimise costs

### 5.3. Why re-invent the wheel?

So we have our basic requirements 'To Do' list, but before we go too far down the road of developing a new application from scratch (and ignoring the learning side of the project for a moment), is there any application which already does the same thing that would save re-inventing the wheel ?

There are astronomy programs out there that are free, but research so far has not found one which provides what I want, most are star locators / databases. If there are any web sites or applications for calculations they only do 'bits and pieces'. I don't want to be jumping from one web site to another or installing too many different applications on my machine.

There is also a lot of repetition on the web (hence the silly numbers in Google hits...) in both information and applications, the idea here is to combine just the best snippets from the web and past astronomy courses in one application and web site.

If an application can be found which complements this one, then it might be used in conjunction with / interfaced to, or listed on the website as a recommended resource.

There may be some repetition necessary, but the whole project is a learning experience as well for the developer and the audience. And adopting an 'Agile' approach we must be prepared to change direction and methods.

## 5.4. How difficult will this be?

The web site and documentation should not be difficult; the main problem is it will be time consuming.

The application itself – making it platform independent and open source will mean it is not tied to one platform and open to a wider audience, however it will mean that it has to be developed with tools that are available on multiple platforms.

The growth of the internet and web based applications show that an internet browser based application is the way forward.

We have no requirement here to store large databases, only perform calculations on small amounts of data. Information on the main objects in our Solar system will be sufficient to cover database development; most of the information should fit in the individual web pages / JavaScript.

At present HTML and JavaScript will be the main 'programming' language, so the project could almost be completed with any text editor.

For most purposes the platforms that Astrolab should run on are Apple OS X, Windows and Linux.

## 5.5. Why develop in Open source + Cross Platform

With open source the fact that the original text of the program code is available to all means it is easier to translate from say PC to Apple even if this is not done by the original author. The fact that the applications are mostly free or very low cost can attract users. There is a wide open source community who may contribute further development or fixes for free.

Support may be intermittent depending on how much time the originator has available and how much community support the project has, but generally if there are a number of open source applications doing the same thing, you would pick the most popular and well supported. Testing an application before using or recommending it is important and we will probably cover this subject in more detail later.

There is unlikely to be a vast amount of money made from this project, but then the tools selected to develop it can be open source so there is little cost apart from time as and when available. **As stated in the Idea Creation (and Basic Requirements) costs are to be kept to a minimum.**



**Open source does not mean that an application is available on multiple platforms – this is known as cross platform development.**

This project will cover both open source and cross platform development, the advantages of cross platform development being that you are not then tied to using one particular machine or even operating system, so you can swap when you choose or as trends change . For example you may be using a PC which may run Windows or it may run Linux, alternatively you may have a recent Apple running Windows or OS X.

## **5.6. Initial Ideas Summary**

As stated in 2.2.2, from current trends and previous experience the actual application should be web based – Web 2 is on its way and developers such as Apple and Google are encouraging complete web based applications. Apple are promoting the development of ‘droplets’, small applications that can run on the new iPod Touch and iPhone, being browser based they should also run on any platform.

Web pages can be built with simple tools down to a text editor like Notepad. Each web browser itself provides most of the interface. There are cross platform programming language tools which can develop code to run in the web pages, but JavaScript is also a powerful tool already built into the browser and does not need any other ‘servers’ running in the background.

Not having to use complex expensive development tools cuts cost and we do not want to be tied to using other products that may be required to run on the web server, or locally, (other than perhaps Java) at the same time to support the application; it should be self contained as far as possible.

Many tools required for this project are available as other open source projects on the web either free and / or supported by donations.

Also we don’t want to become too reliant on always having a web connection, but again if the application is self contained then web browser based applications can be run from a local drive on a PC or PDA.

On the subject of low cost / free tools check out <http://sourceforge.net/index.php> where there is a vast array of applications available.

**Appendix C** contains the initial ‘To do’ list for the Astrolab app part of the project, produced while doing the initial investigations. **Appendix D** contains the ‘To do’ list for developing the supporting web site for Astrolab. The two can be done in parallel as they are independent.

## 6. Current Astrolab Version is 0.9864\_20140325

Astrolab has been following the accepted version number convention, where version 1.0 is accepted to be the first fully working release. However this really means nothing in the real world, version 0.981 could be almost complete and never need much change or version 4.6 could be considered complete, in that what is in it works, but users find it lacks useful functionality for its intended purpose.

There is a lot of open source software out there that has high version numbers, but has not been updated for a long time, either through lack of time, interest or the developer has moved on. In that case you cannot be sure how much longer it will be supported, no matter how good the application is.

So for an indication of support and continuing development we include a date in our version number 0.9864\_20140325. While in development the date is the intended final release date for the current iteration being worked on, so the version number may still increment until the release is put on Source Forge.

[updated 20140402]

## 7. Installation

For the Windows version – Unzip the complete folder and place anywhere, the application is self contained and web browser based.

The OSX version can use Apple's own tools to package the application ready to copy to a disc image type file that can be downloaded and installed via their own install process into the usual Applications folder.

Or it can simply be compressed to a Zip file and the folder unzipped in OS X and the folder placed anywhere convenient to the user.

[ Move to User GUiide]

## 8. Starting the Application

Navigation is via normal navigation for the web browser in use. To start the application click on any of the three html pages listed below, or you could rename "auto.html" to index.html if Astrolab is run as a subfolder under your own web server. "auto.html" attempts to select the most appropriate framework based on the OS it detects.

**mMain.html** starts 'mobile device version' – reduced graphics, simple menus, each page displayed in full, no frames

**main.html** starts full version in frames

**auto.html** attempts to detect 'device' to load appropriate version

fDefault.css and fEnvironment.js contain parameters to set font size and layout, a mobile device / tablet version usually being set to have a smaller default font.

## 9. Application Contents

### **AstroEquations.xls**

Equations, constants and values used in the Astrolab application. Spreadsheet AstroEquations.xls contains a list and ‘test harness examples’ of equations to be used in Astrolab. It also lists values for constants and measurements used in Astrophysics, using National Physical Laboratory and CODATA recommended values.

## 10. Develop

Originally there was a 'tablet' based version and a 'windowed' version for desktop use. The principle was that the core of the application was identical in both the local PC / Tablet and online / standalone application. This made use of HTML frames for a 'windowed' version, however this increased the work required, so now the 'tablet' version has been adopted for wherever the app is run. All options are now controlled by a simple menu selection system on main menu pages and individual function pages

The mMainTab.html page starts the application.

All core applications have file names starting with 'c'. Application framework files that run the core pages have file names starting with 'f'. HTML file names follow the Java function naming convention of starting with a lower case letter e.g. planetData.html

Application 'windows', menus and calculations are driven by HTML 5 and Javascript.

The background colour code in the header of Astrolab pages shows current state of development of the individual page, and is also shown on some sections of application pages, supporting web pages, documents and application status tables etc, to show status of some individual items.

Development / Prototype (html bgcolor #FF00FF)	Testing (html #FFFFCC)	Completed (html #CCFFCC)
---	---------------------------	-----------------------------

[updated 20140408]

## Development Work Flow

Below is a simple list showing how development and files should flow through the various stages of this project and the development, test and live (user) environments.

<b>Ideas / Enhancements</b>	>>	sourceforge tracker list	>>	web site specification	
			>>	application specification	
			>>	blog news	
web site specification	>>	web site dev	>>	web site test	>> web site live env
application specification	>>	app dev env	>>	app test	>> app live env
	>>	app user guide }			
		app readme.txt }			
		app change log }	>>	app test	>> app live env

## Bugs

>> sourceforge tracker list	>>	web site dev	>>	website test	>>	web site live env	>>	sf tracker
				app dev env }				
				app change log }	>>	app test	>>	app live env
								>> sf tracker

## News

>> sourceforge.net	>>	web site as appropriate
	>>	blogger.com
	>>	twitter.com

## 11. Astrolab Application Build for Distribution Process

There are few programs to compile at present as the whole application is based on HTML and JavaScript, some Java may creep in later.

All that is required to copy a development release from the development folder to a test or live folder is to:

Clear the destination folder if appropriate, i.e. some files are no longer required and any link to these redundant files may be lost or the files will simply take up space.

If all the files in the new release are either existing files or new files then they will simply overwrite the old versions / create new files when copied.

Check / update the fEnvironment.js file with the version number of the current release to be distributed.

Update the cDocuments\Readme.txt file with changes and version number for this release.

In this case the 'folders' are the 'Sites' folder under ~User in OS X – this is used for development and testing – and the [www.alchymy.com](http://www.alchymy.com) 'Projects' folder on the on line web site. Online is live and iMac is development and test, there are no multiple versions in development and test stages as we had originally when development was done under a Windows XP platform.

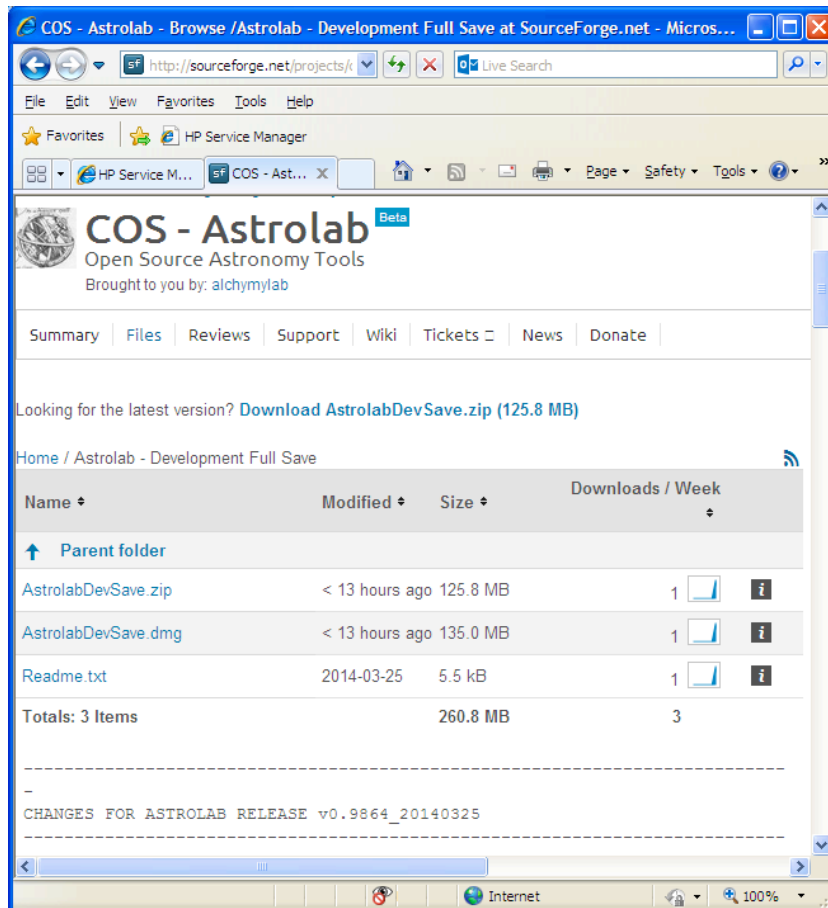
Once tested, any change is simply uploaded to the folders on the alchymy.com web server site, so the on-line version evolves.

A script is run to build the 'package' for both backups and final releases.

It takes the 'Sites/Projects/Astrolab' folder under '~User' and copies it to the '~User/Uploads' folder (a 'clone' of the 'Downloads' folder with the arrow reversed!). A copy 'Astrolab' folder is created under 'Uploads' and the 'Readme.txt' file under 'Astrolab/cDocuments' is copied out to 'Uploads'. The Astrolab folder is a full version with all Datasets and 'addin's'.

The script then creates an AstrolabDevSave.zip file and a compressed AstrolabDevSave.dmg file. Both can then be loaded to the project specific 'sf.net' site files section for Astrolab, together with the 'Readme.txt' file. This is currently done using FileZilla for OS X.

'Official Releases' other than development saves are then built with the same script, and the '.zip' and '.dmg' files simply renamed as Astrolabv0\_9864.zip or Astrolabv0\_9864.dmg before uploading to 'sf.net' folder 'Astrolab Latest Release'.



, and zipped copies of the Astrolab application folder (it can all run stand alone) uploaded to sf.net Conquest of Space Astrolab project.

Data files – if amended – are uploaded as a separate package as they take up space and are not directly linked into the app at this time. So the ‘Datasets’ folder can be removed from Astrolab to save space, and a later set downloaded separately. Some modular ‘addins’ are also stored separately on sf.net so you can just download the core application on its own.

Build package on Mac with Package builder and then add to a disc image for upload to sourceforge.net

Twitter (mainly), Sourceforge.net project news and Facebook are used to ‘present’ new versions on the net and so indicate a new release is now available.

[updated 20140410]





## 12. Phase – Planning

### 12.1. Stage – Scope

The project will cover the entire process from:

The COS Project can be split into two main sections

- The supporting website – which will cover
  - i) An associated blog which acts as an initial notepad which may then have items expanded on the main web site and also mirrors the main web site links to Astro resources
  - ii) Link to a sourceforge.net web page to explore sourceforge.net facilities in supporting the project
  - iii) Astrolab application to support calculations & development of that application
  - iv) creating the hardware platform
  - v) installing the operating system
  - vi) selecting the software to support the hardware, the website, the application development, and maintain the system
  - vii) developing the website and application
  - viii) documenting of the whole process
  - ix) on going maintenance and development
- The application – Astrolab

The application is intended to provide a tool to cover common astrophysics related data, laws and calculations which can be used on the web site or independently of the web site on any platform. As described in 2.1 Astronomy, Astrophysics and Cosmology are referred to under one heading of 'Astrophysics' for the purposes of this project.

In order to keep the initial amount of work down and keep the project start up under control it will initially only cover the items listed below which will help prove the concept:

- i) Data on planets and the Solar system
- ii) Astrophysics calculator
- iii) Distance and temperature conversions
- iv) A Glossary of common terms used in Astrophysics
- v) Initial development will be for PDA with a desktop / standalone / online version derived from this

Using the Iterative development model further functions can be added later once the initial development is proved to work.

## **13. Phase – Requirements**

### **13.1. Stage – Specification for Platforms used in Development**

#### **Development platform – hardware, basic**

iMac 21" / OS X Mavericks

#### **Development platform – software**

#### **Hosting for Website**

Website – UK2.net – conquestofspace.com

Blog – blogger.com – <http://conquestofspace.blogspot.com/>

Project – website and sourceforge.net

Email – alchymylab@gmail.com

## **13.2. Stage – Detailed Specification – Supporting Web Site**

## **Phase – Analysis and Design**

### **14. Phase – Implementation**

The whole project is based around HTML pages and Javascript – all text files, designed using various HTML tools. No programs need to be compiled in current versions. Zipped folders are produced for deployment on platforms other than the development machines.

### **15. Phase – Deployment**

To install website files they can simply be run from any folder they are stored in, or for Live releases they are simply uploaded to the web server.

The application files are also self contained, the zipped application is simply downloaded from the Sourceforge.net project page and copied and unzipped to the desired platform and the application started by loading the main.html page in a web browser.

### **16. Phase – Testing**

Final releases will be tested and by the developer(s) and users. Any bugs will be recorded on the Sourceforge.net project page. Any bug fix will be recorded on Sourceforge.net list and may be included in the AstroChgLog.txt file with the next version made available for download on sourceforge.net

### **17. Phase – Evaluation**

Any issues raised from testing or enhancement ideas should be logged on the Sourceforge.net project page and may be incorporated in a specification for the next iterative / incremental development and release, if a large development, or simply listed in AstroChgLog.txt file if changes are included with the next version made available for download on sourceforge.net.

## 18. Appendix A – Initial Ideas - ‘To Do’ List for COS Astrolab

<b>Items Independent of Application Functionality</b>
Must be platform independent - so automatic support under MS Windows, Mac OSX and Linux
Core is based on HTML standard and meets W3C standard
Must meet accessibility standards
Can be run from a Web server or standalone
Standalone installation must be a simple process – drag and drop application folder
Uninstall must be a simple process – delete application folder
Programming, code must be understandable, well documented and platform independent – (Javascript only at present)
All new releases to be cumulative
Meaningful version identification to be used in all application versions and supporting website conquestofspace.org
<b>Application Interface:</b>
Web Browser
Possible W3C compliant
Simple frames to avoid duplication on individual pages
<b>Content:</b>
Initial development will be for PDA with a desktop / standalone / online version derived from this
Astrophysical Constant List
Astrophysics calculator
Convert Distances
Temperature conversions
Equations - Drake Equation
Glossary of common terms used in Astrophysics
Mathematical Constants
Periodic Table – List only
Physical Constants
Solar system data
Units of Measurement
Equations - Sphere volume and area calculation
Third party resources used as part of the app:
None at present
<b>Facilities:</b>
There will be a download facility for a packaged version of the main application under the supporting web site

## 19. Appendix B – Initial Ideas - ‘To Do’ List for COS Website

Web support for the application will split into 3 parts

- 1) A blog which acts as a scratchpad for news, views etc which may then be expanded upon in the main web site pages if relevant. All links to relevant web sites and news / notes appear on the blog first. Links are then added to the main web page(s). There is a latest news item link on the main web page which points to the blog.
- 2) A sourceforge.net account to promote the application, supporting web site, and act as a store for downloads, source files and supporting documentation.
- 3) The main web site with more detailed information on the application and how it and the other components are developed and supported, and expanding on the blog records if relevant:

The standards below should be used for the main web site:

- XHTML is now a widely used standard and all pages should meet the W3C XHTML 1.0 Transitional design standard as a minimum. However a switch to HTML 5 should be considered.
- The sitemap should be one of the first pages created from the template and should act as a simple database to list all other pages on the web site. It can be used to sketch the basic layout of the site by subject / title. The actual pages themselves and their content are created later.
- The sitemap content should list every other page and its links other than the standard menu bar links, which should be shown for the home page only.
- Website should be based around a template page(s) so all the pages are standardised for easy maintenance. Any page that is a template should have a name ending in “\_T.html”. Any design changes common to all related pages should be applied to these pages first, then all other related pages.
- Basically pages should have a header section, menu bar, content section and footer section which can be created as 4 separate tables spaced apart vertically.
- The Header and footer should rarely change, the footer should indicate when each page was last updated.
- The content section is where most changes should occur.
- To speed up navigation from each page a simple common menu bar should be included under each page header. It should only contain the most frequently used links and a home page link.
- Infrequently used links should only be shown on the home page or on a relevant content page.
- No pop up menus, windows or advertisements. No animated graphics. Standard text box Google content related advertisement is allowed.