

# Warp2010 – Leonov Edition

## WHAT IS WARP 2010?

Warp 2010 is an utility geared towards 2300AD players. Its main purpose is to help them calculate jump routes between a star and another in the Near Star Map, and generate solar systems for these stars, if needed. In the beginning it was just a simple mapping and route calculation tool – Warp stands for “Warp Assistant Routing Program” – but in the last year it has grown increasingly sophisticated.

First edition of this program saw the light in the late nineties: it was written in Delphi 2.0<sup>1</sup>, used to run on a 16-meg ‘486 running windows-95 and, to my knowledge, it is still used in 2300AD community. This version was developed in my spare time until 2000, when I lost interest in maintaining it.

I restarted working on it in 2009, when I wanted to use my solsys generation routines to generate Celestia (<http://www.shatters.net/celestia>) system definition. Old habits never die, and I found myself translating old Delphi code into C#. Then, on Celestia forum, some open-source fanatic challenged me into writing it in a “more open language” since he stated that C# was the work of the Dev...ehm, “tied to a single vendor and not portable at all”. I said that, thanks to Mono, porting Winform apps would be no issue at all, and C# as it is was an ECMA standard and therefore open by definition.

Then I tested a tentative pre-alpha release on linux using mono runtime: the program did start, but it was extremely slow, very hungry on resources and crashed frequently. I wanted new Warp edition to be multiplatform, and these tests disappointed me a lot. Moreover, I did not want to use an interpreted language<sup>2</sup>: Warp 98 was lean and mean, and I wanted Warp2010 to be the same AND multiplatform.

Then I checked out QT from Trolltech (now Nokia) and I saw the light. Great licensing options (GPL and LGPL), a great package for free (IDE+QT Libraries +MingW C++) and a powerful framework to lean upon, with a C++ coding style more akin to Delphi programming than so-called “Modern C++ Programming”. And last, but not least, a truly multiplatform library (the C++ code compiles, with no modifications at all, on Windows, OSX and Linux) .

So I bit the bullet and learnt C++ with QT. And while there are some things I don’t like about C++ (lengthy compile cycles, syntax quite obscure here and there) I now think that QT/C++ is the only way to write native, multiplatform desktop applications.

But I digress. With the help of QT, Warp did gain a lot of new features, not available in earlier editions. A REAL map to browse and navigate. PDF export. Hexmaps. Latest feature I

---

<sup>1</sup> Great, GREAT programming tool and awesome programming language. If you have money to spend, want to build blazingly fast apps for Win32/64 environment, Delphi is The One True Way.

<sup>2</sup> Granted, c# is not a real compiled language, since the compiler translates C# code into an intermediate form.

implemented was a Diaspora Cluster Generator. Yes, Diaspora and 2300AD are different games, with different backgrounds and coming from different eras. But I wanted a Cluster Generator to live side by side with a Star Sector mapping tool and a Solar System generation tool. And this is what I coded.

## WHAT IS 2300AD?

2300AD (*né* Traveller: 2300) was a hard-scifi RPG published by the now-defunct GDW in 1986. While it featured FTL starships it was quite unique at that time because it tried to promote a “realistic” sci-fi feel. For instance, there were no starfaring empires: instead, nation states explored and colonized remote worlds, and it was devoid of all techno-magical stuff often found in sci-fi rpgs of the time.

Its most distinct feature was the Near Star Map: a poster-sized map of the nearest stars created using the Gliese 1969 star catalogue, which was the most accurate reference at the time. That very reference was included in the game package.

The game also put limits to FTL jumps, stating that a starship could only travel faster than light for no farther than 7.7 ly and the extremes of FTL flight could only take place in a significant gravity well; that is, in a solar system. These limits dictated the formation of exploration routes – the “Arms” in 2300AD parlance – and the emergence of stellar crossroads – star whose position would imply a strategic importance.

Since the “Arms” depicted in the background did cover only a part of the possible exploration path, leaving to the player the not-so-enviable task to calculate all possible routes. That would lead, some ten years later, to the development of Warp.

## WHAT IS DIASPORA?

*Diaspora* is a hard-scifi based upon FATE ruleset. Its most remarkable feature is that there is no ready game background to lean upon – you get to *build* it in your first game session. Boring? Hell, no! It’s really, really, REALLY fun! Central to this task is the concept of *star cluster*: players create the stars that make it up, determine star linking routes and assign them aspects – descriptive tags that remark that summarize the system in three short phrases.

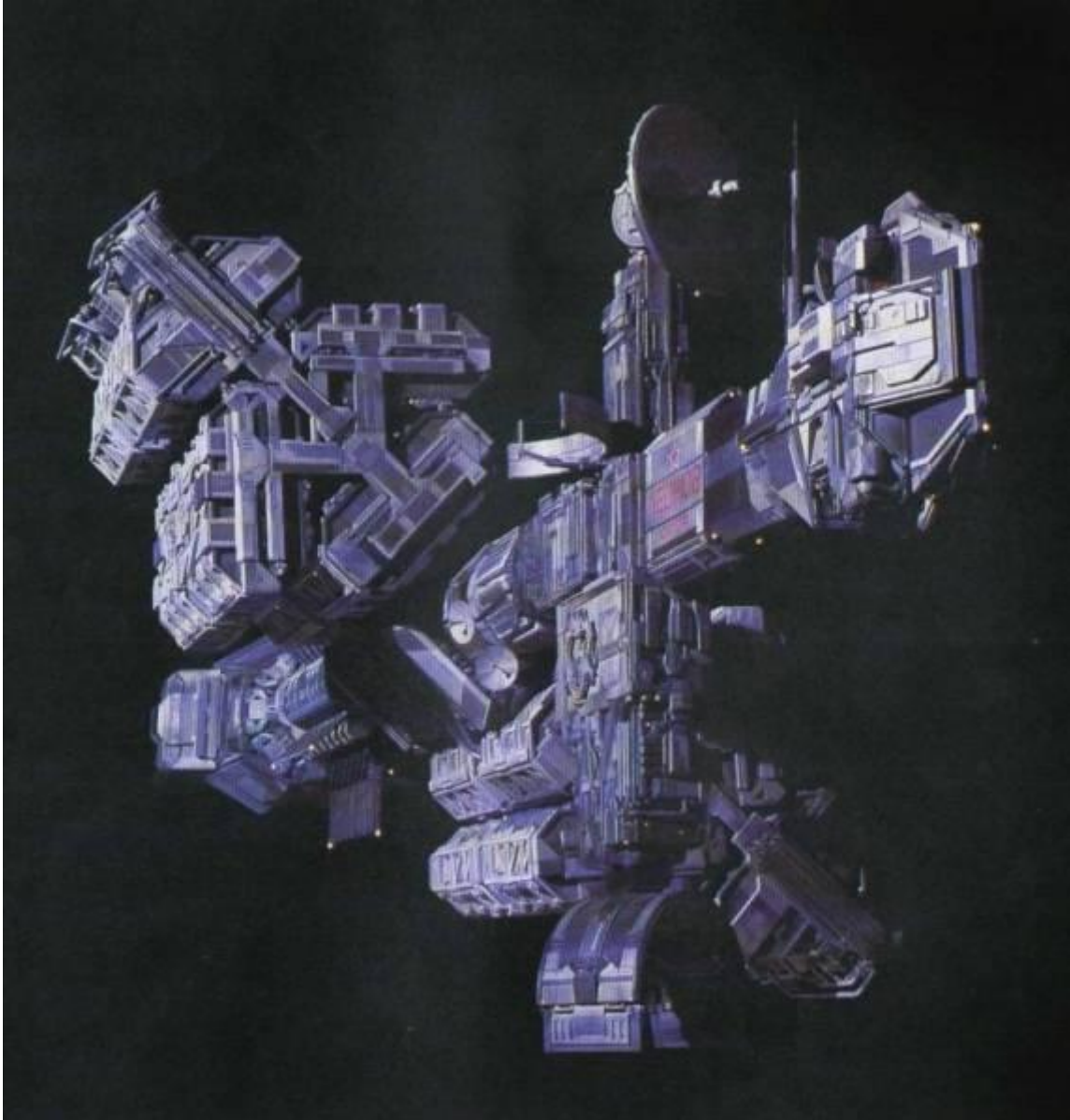
There are other reasons that make Diaspora a great game – the FATE Engine, its mini-games, its writing style. But the feature that, as a grizzled gamemaster, won me over was cluster building. Simple, elegant and powerful.

## WHY MERGE SUCH AN UNRELATED FEATURE AS CLUSTER GENERATION INTO WARP?

Because Warp is MY Star Routing Program and I wanted it this way. Besides, I plan to merge Cluster Generation with Star Sector and Solar System generation. That way I could better detail a cluster, and while Diaspora does not require this level of *color*, it is a “nice to have” thing in my book.

## WHY “LEONOV EDITION?”

Because the “Leonov” spaceship of “2010 – The Year We Made Contact” movie is the perfect 2300AD ship: artificial gravity provided by rotating habitats, a rugged anaereodynamic look and worn-out interiors. Incidentally, it is also the perfect Diaspora ship. So, since it is 2010, I chose this name to christen my program.



(Why don't we have these babies flying in space now? Why? Why? It's 2010, dammit!)

# First Contact

## SYSTEM REQUIREMENTS

The program runs reasonably well on these minimum resource requirements:

### WINDOWS

- Windows 2000, XP, Vista or 7. Can't say anything on Win98 or WinME (yuck!)
- 128 Megs of RAM (Tested on 2000 and XP with this amount of RAM – Virtual Box Rulez ☺ )
- 15 meg of hard disk space.

### MAC OSX

First and foremost, you need an Intel-based Mac. Moreover, since OSX is not my native platform, I could not test as extensively as I did for Windows.

The platform I had access to was a 2gig, T2300 MacBook pro from 2006, running osx 10.5.2. I could not test on newer releases of OSX, but I expect Warp2010 to run on Mac with OSX from version 10.4 onwards and at least 512 meg or RAM.

Disk requirements are slightly higher – about 38 meg of RAM with QT libraries.

If you are a technical type, have an aging PowerPC machine and don't mind compiling some C++ source, you can grab QT from <http://qt.nokia.com>, get the source and try building Warp for your machine.

### LINUX

Warp2010 **runs** on linux but I have to admit: I didn't build a binary distribution for any linux distro yet. I tested it on Ubuntu 9.10 and a VirtualBox virtual machine running Xubuntu 10.4 with half a gig of RAM. On both platform Warp compiles and run almost flawlessly. I cannot speak for other Linuces, though.

## INSTALLING WARP

### WINDOWS

Unzip the package to a new “Warp2010” folder, click on the “Warp2010” icon and you're set. Yes, there's no installation routine. It's a “Portable App” in its own right. Uninstalling the program is also this easy: just delete the Warp2010 folder to remove the program from your system.

### OSX

Almost the same as Windows: unzip the package in a folder, browse the newly created one and click on the Warp2010 icon.

## LINUX

As per today, there's no .deb or .rpm package. I plan to create a Debian package sometime soon for easing the installation process as much as I can. On a plain vanilla xubuntu 10.4 install, the only dependencies needed by Warp2010 are libQtCore-4 and libQtGui-4, (check out for version 4.6 onwards: older Qt version don't work)

If you don't feel like compiling Warp2010 from source, you can download Warp for win32 and run it by means of Wine emulator. It's not as fast as a native linux version, but it's quite usable in all of its star routing glory as well.


## BUILDING FROM SOURCE – SOME ADVICE FOR THE DEVELOPERS

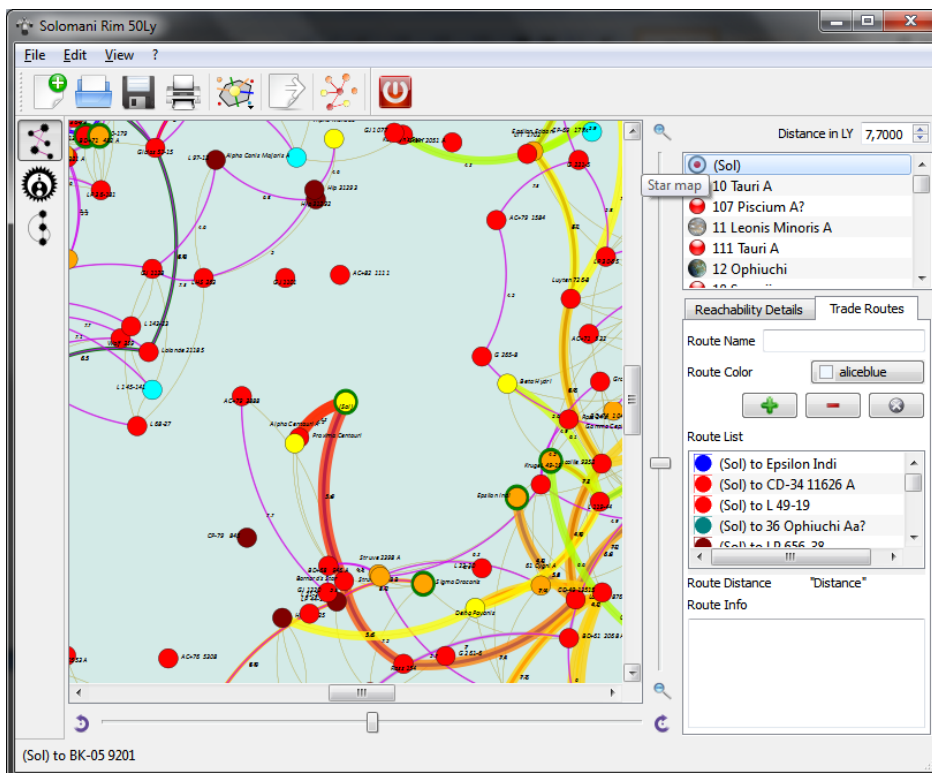
If you belong to the coders' club, here's some advice for building the whole stuff from scratch:


1. If you want to start easy, get QtCreator package from Trolltech: <http://qt.nokia.com> . Be warned, this is a huge download even if you have hi-speed internet. (ranging from 200 to 400 meg)
2. If you're familiar with makefiles and want to use your favorite programming environment (emacs/g++, or MSVC or XCode) you can just download only the libraries.
3. Then, you can get the source using SVN from this address:  
<http://subversion.assembla.com/svn/qtwarp2010>
4. Once you've got the source, you can open the project using QtCreator if you're the IDE type, or you can qmake-it out to generate Makefiles, then compile it.
5. When compiling is over, you can head to the executable file and just run it. It will create the datafile it needs on ~/warp2010\_data directory.
6. You're done!

# WARPing thru the uncharted space

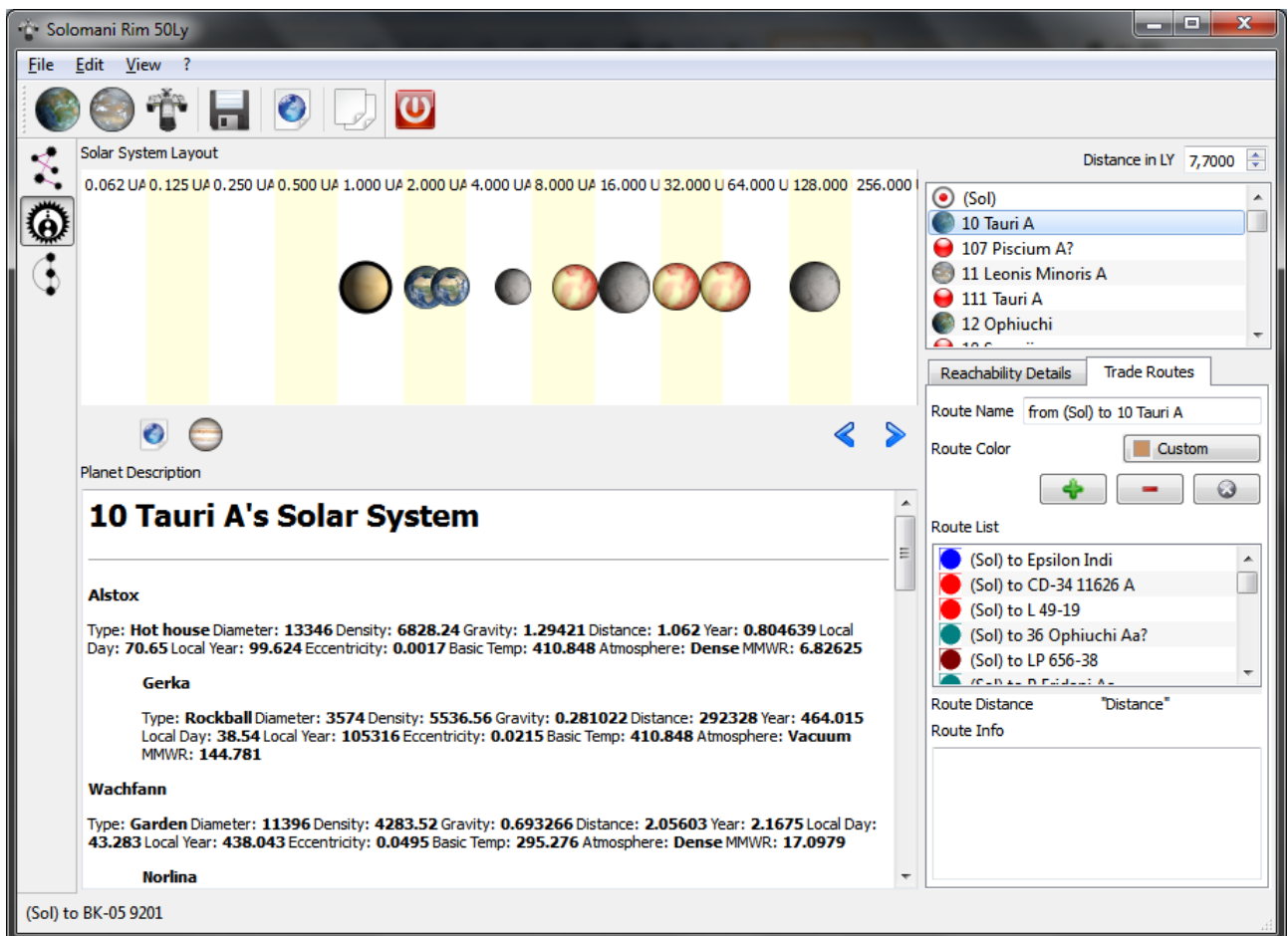
The Program is divided in three main areas:


**Star Sector Area**, where you create, load and save star sector, select the reference star whence jumproutes originate, change jump limit, switch between sector visualization modes and other things. You access this area by clicking on this icon  on the left pane.

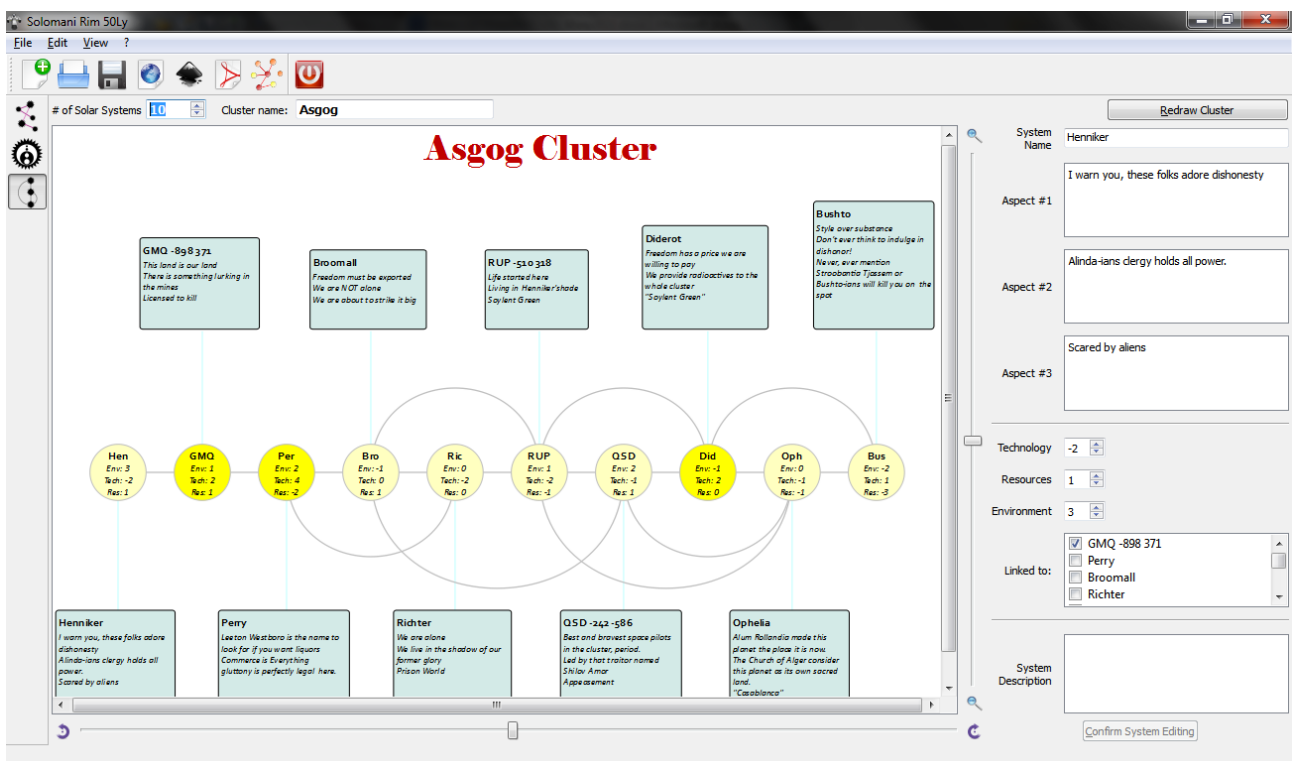


**Solar System Area**, where you create solar systems and trade routes, and inspect a star's solar system. Clicking on this icon  lets you access this area.






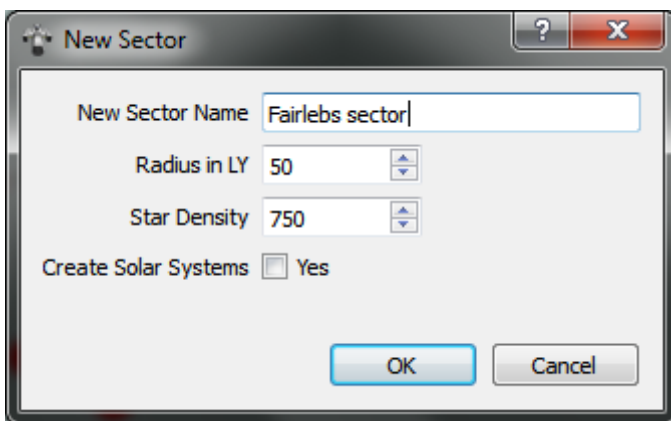
**Diaspora Cluster Area**, where you can create, edit, view, export and print Diaspora Clusters. Again, clicking on this icon  lets you access to Cluster Area.



# Star Sector Area – Operations


## CREATING A STAR SECTOR

You create a brand new star sector by clicking on this icon . A dialog box then appears, letting you specify the name of the sector to create and whether you want to have solar system and trade routes created along the star sector.



The system presents you a default, randomized name. You can also specify the radius in light years of your sector, and the star density of the default cluster. Obviously, the stars created with this method are completely fictitious. However, if you want something more realistic, I provided three star sector files based upon realistic data.

## LOADING A STAR SECTOR

To use these files, you must load them in Warp2010. You can load a Star file by clicking on this button  in the Star Sector area.

Three sector files are provided:

- **2300AD.STR**, a star file containing 2300AD star definition
- **50ly.STR**, a star file containing the nearest stars in a 50ly sphere as drawn from the Hipparcos '98 star catalog.
- **100ly.STR**, a star file containing the nearest stars in a 100ly sphere as drawn from the Hipparcos '98 star catalog.

## STAR ROUTES

Once you have fed Warp2010 with star data, the system will draw a reachability map from the reference star according with your selected jump limit (the default is 7.7ly, which is 2300AD limit, but you can change it to your heart's pleasure).

There are three types of route that get drawn:



## OPTIMAL PATHS

These are the shortest paths connecting the reference star to any other reachable stars. These routes are color-coded, starting from red and ending in violet through all of the rainbow's colors. The longest the route, the more complete the spectrum. This means that short paths will start in red and end in orange, yellow; medium-length ones will end in green and longest one will end in violet.

## SECONDARY PATHS

These are the paths connecting a reachable star to its neighbor. These are thinner than optimal paths and grey in color.

## TRADE ROUTES

These are special user-created paths that connect the reference star to a given reachable star. You can create Trade Routes in three ways:

1. By forcing their creation when loading or creating a star sector: this option forces the creation of a trade route for every reachable star containing an inhabitable planet
2. By forcing their creation in a similar way, from the Solar system area
3. By selecting a reachable star from the star list and creating one. As depicted in this example, when you select the star you are presented a default name. If you wish to add this route to your trade route list, you choose the route color and press the "+" button.

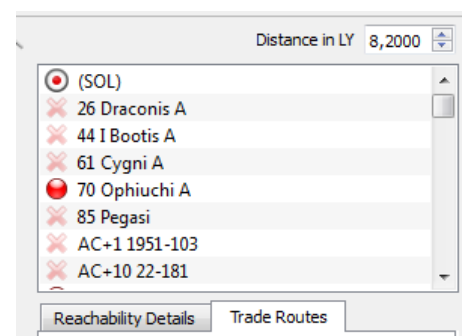
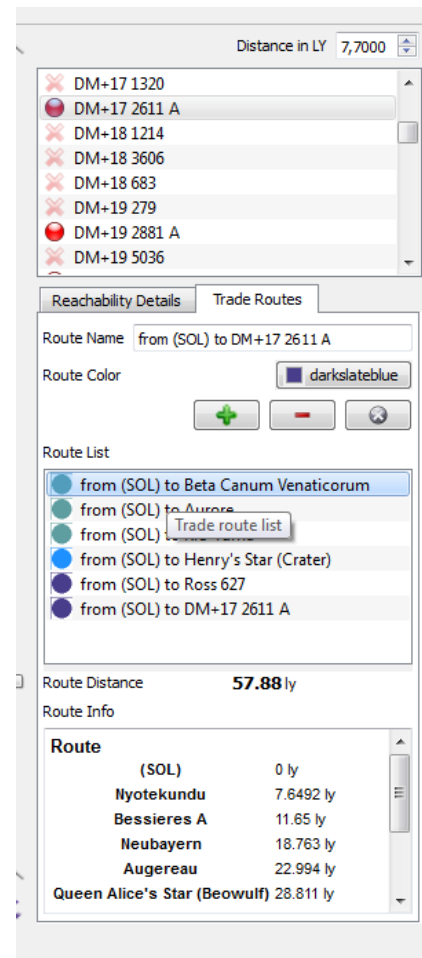
Trade Routes help you organize your star map by highlighting your "Arms". This feature is not a mere duplicate of "Optimal Path", since there is a sector display mode that shows only the user-defined Trade Routes.

## CHANGING REFERENCE STAR AND JUMP LIMIT

You are not limited with a single jump limit or a given reference star. You can change both of them.

If you wish to change the distance in LY, you've got to operate on the input box labelled "Distance in ly". However, to make you change effective, you have to double-click on the reference star (identified by a bull's eye icon).

If you want to set a new star reference, you must double-click on your chosen star. The whole starmap will change according to your new settings.



## MAP DISPLAY OPTIONS

Once you've loaded a star sector, you can change the way it is displayed. There are three display modes to chose from. These are:

### NORMAL DISPLAY MODE

A normal projection on the x-y axis of the star map. Z-index is rendered by drawing smaller or bigger star circles.

### HEXMAP DISPLAY MODE

Stars are rendered on an hex grid (this mode will be familiar to Traveller players). Overlapping stars, that on the Normal Display Mode share (more or less) the same position are arranged in free hexes. This map, while not being overly "realistic", can be easier to read, since stars are somehow uncluttered. This leads us to...

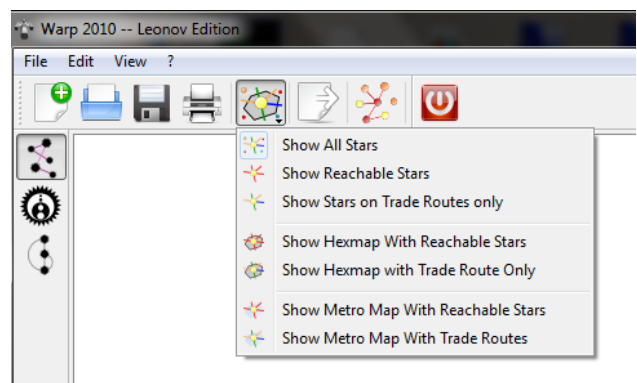
### "METRO-MAP" DISPLAY MODE

Which is nothing more than Hexmap mode without the hexes. I call it "Metro Map" because what I wanted was to automatically generate a simplified starmap resembling those found in subways all over the world. There are no easy to implement algorithms, so the best I could find was using the Hexmap uncluttering feature and removing the hexes from the resulting map.


### CHOOSING A DISPLAY MODE

You choose a display mode by clicking on this button and waiting for about one second (600ms, actually). Then a menu pops out, allowing you to select from the following options:

- **Show All Stars** – Show a normal star map with all stars, reachable or not, all trade routes and optimal paths
- **Show Reachable Stars** -- as above without trade routes
- **Show Stars on Trade Routes only** – as above, with trade routes only
- **Show Hexmap with Reachable Stars** -- Shows an hexmap with reachable stars only
- **Show Hexmap with Trade Routes only** – as above, with trade routes only
- **Show Metro Map with Reachable Stars** -- Shows a metro map with reachable stars only
- **Show Metro Map with Trade Routes only** – as above, with trade routes only.




## SAVING A STAR SECTOR

Once in the Star Sector area, press on the  icon.

You can choose from two formats:

- A simple star directory (.STR), which is a text file containing star informations
- A file containing not only star informations, but also saving solar systems and reachability data. However, this data is saved in a binary format, and its extension is .starx.

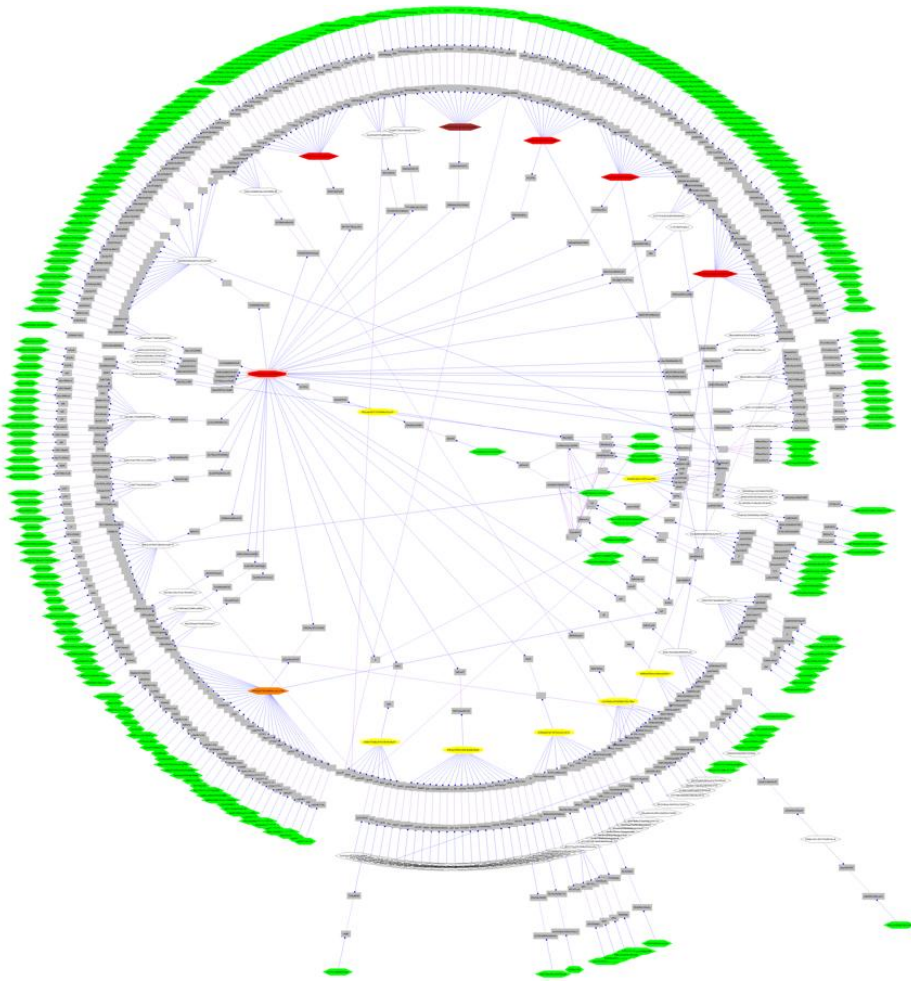
## EXPORTING A STAR SECTOR

By clicking on the  icon, you can export your map in Adobe PDF or SVG. Exporting in SVG allows you to edit the resulting star sector map in graphics program like Adobe Illustrator or Inkscape (<http://www.inkscape.org>).

When exporting in PDF, you will be asked whether you want a single, huge PDF file or a paginated, atlas-like, PDF document.

### EXPORTING IN GRAPHVIZ

GraphViz (<http://www.graphviz.org>) is a tool used to automatically draw graphs. You define graphs by using a description language, feed this data file to a command line tool and what you get is an automagically generated graph. I reckon that such a feature is more useful when generating Diaspora clusters (you can create GraphViz files out of Diaspora Clusters, too) but hey, I wanted to see if I could automatically generate something like this 😊




# Solar System Area – Operations

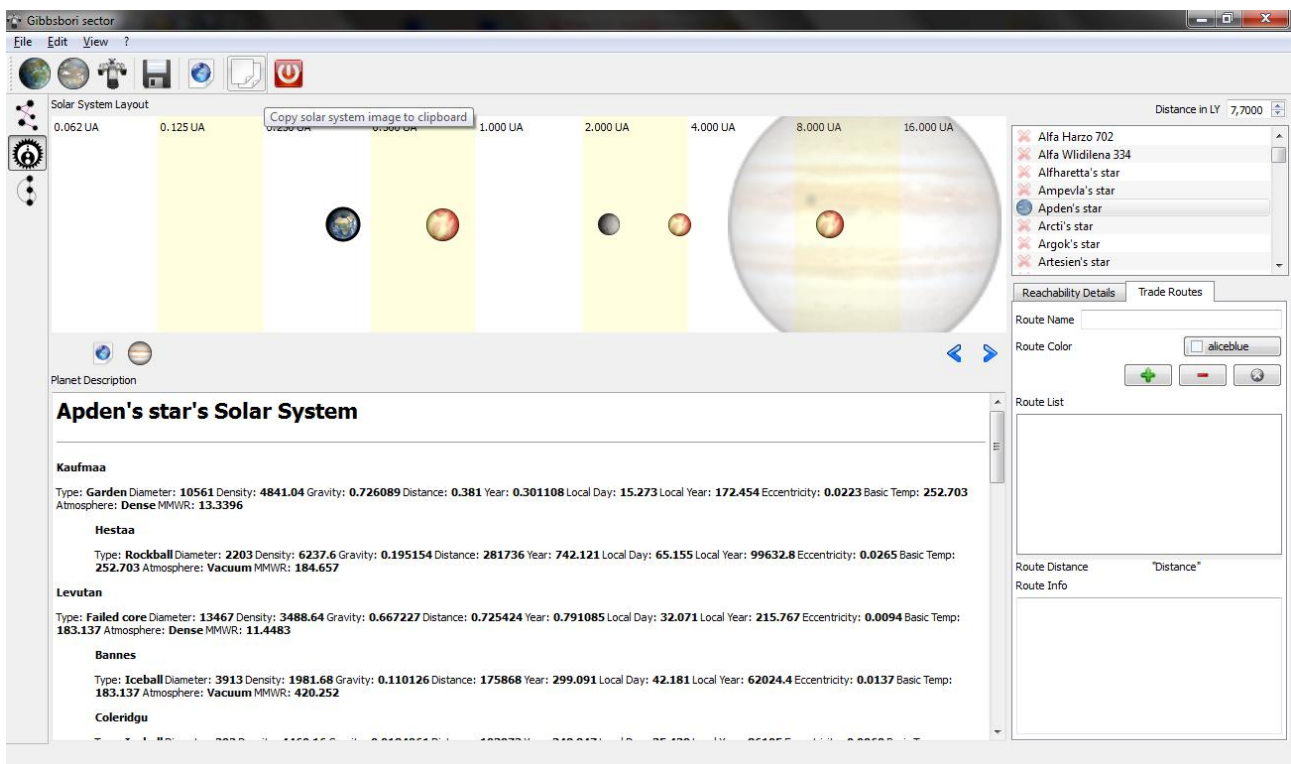
## A WORD OF WARNING:

The Solar System Area is, in some ways, quite obsolete. While Star Sector View and Diaspora Cluster View currently benefit from an advanced visualization framework (you can zoom in/out and rotate the view, for instance) the Solar System view is less interactive, with less options.

Eventually, I'll bring Solar System Area on par with other areas, functionality-wise. But for now, you'll have to make do with a simple solar system visualization widget.


## CREATING A SOLAR SYSTEM

First, you have to create or load a Star Sector (duh). Then, you can choose to create a solar system for a selected star. In this case, you select the star you want to populate on the star list on the left, and then press the  icon. According to star's size and color, a solar system will be created.




Clicking on a star containing a solar system updates the view; on the top, there is a logarithmic map of the planets in the system; on the bottom, a textbox containing basic planetary info. (Again, something sort of incomplete).

## CREATING SOLAR SYSTEMS FOR ALL STARS IN THE SECTOR

Since there can be over 5,000 stars in a star sector, and since computers are especially good at doing repetitive tasks fast, we can choose to force solar system creation for all stars in the list by clicking on  icon.

## CREATING TRADE ROUTES

Now that we've populated our star sector with solar systems, we might want to establish preferential routes to all systems containing garden planets. By clicking on this icon  we force the creation of trade routes for all star containing at least a marginal planet (that is, a garden planet that's not so earthlike) .

## EXPORTING A SOLAR SYSTEM

We can export Solar System data in a number of formats:

### CELESTIA

(Needs work. What the routine does is creating an SSC (Celestia Solar system file) containing all planetary data).

### HTML

Dumps to file the contents of the textual description pane of the system.

### CLIPBOARD

Copies the solar system map in the clipboard (Windows only).


## FUTURE DEVELOPMENTS – BUT DON'T HOLD YOUR BREATH.

- Creating a more interactive solar system view, containing satellites too.
- Adding a planet editor, to allow editing of a single planetary body of the system.
- Export in SVG, PDF.
- Better integration with other two components – Star Sector Area and Diaspora Cluster Area.

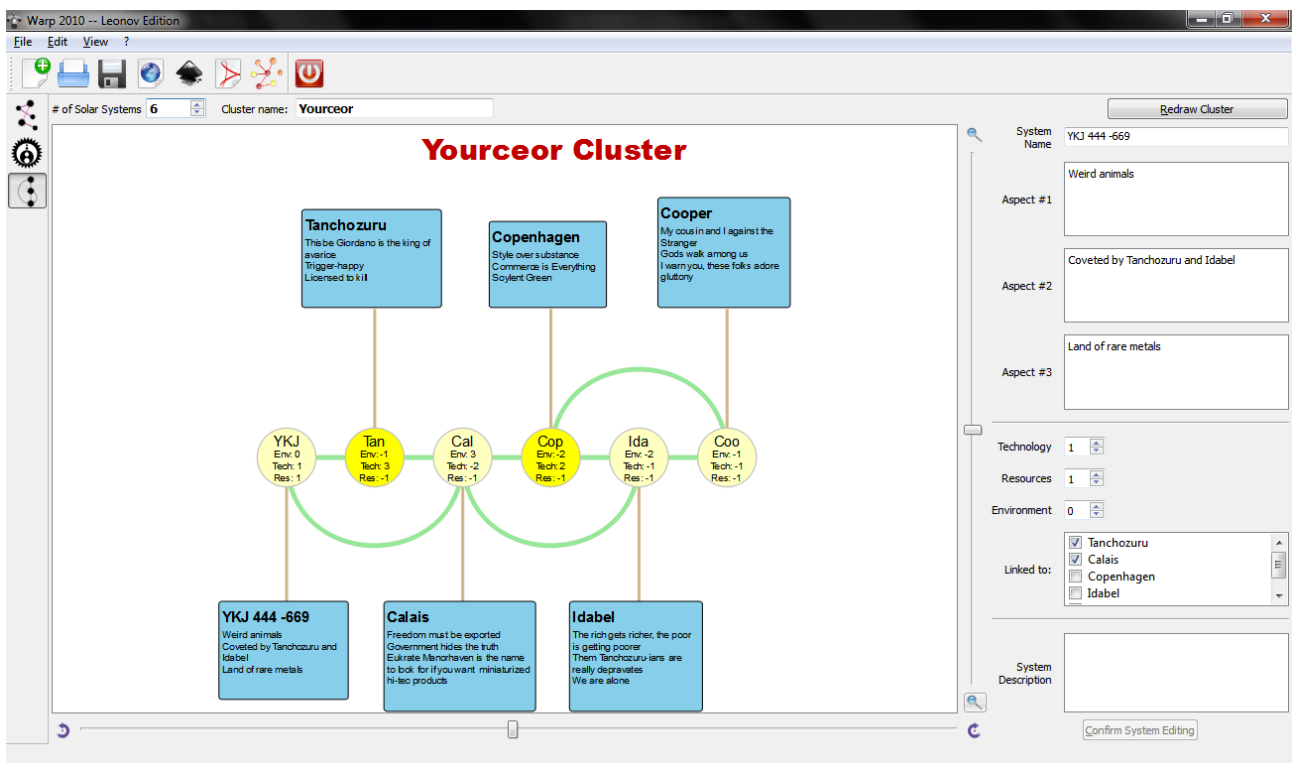


# Diaspora Cluster Area – Operations

## CREATING A CLUSTER

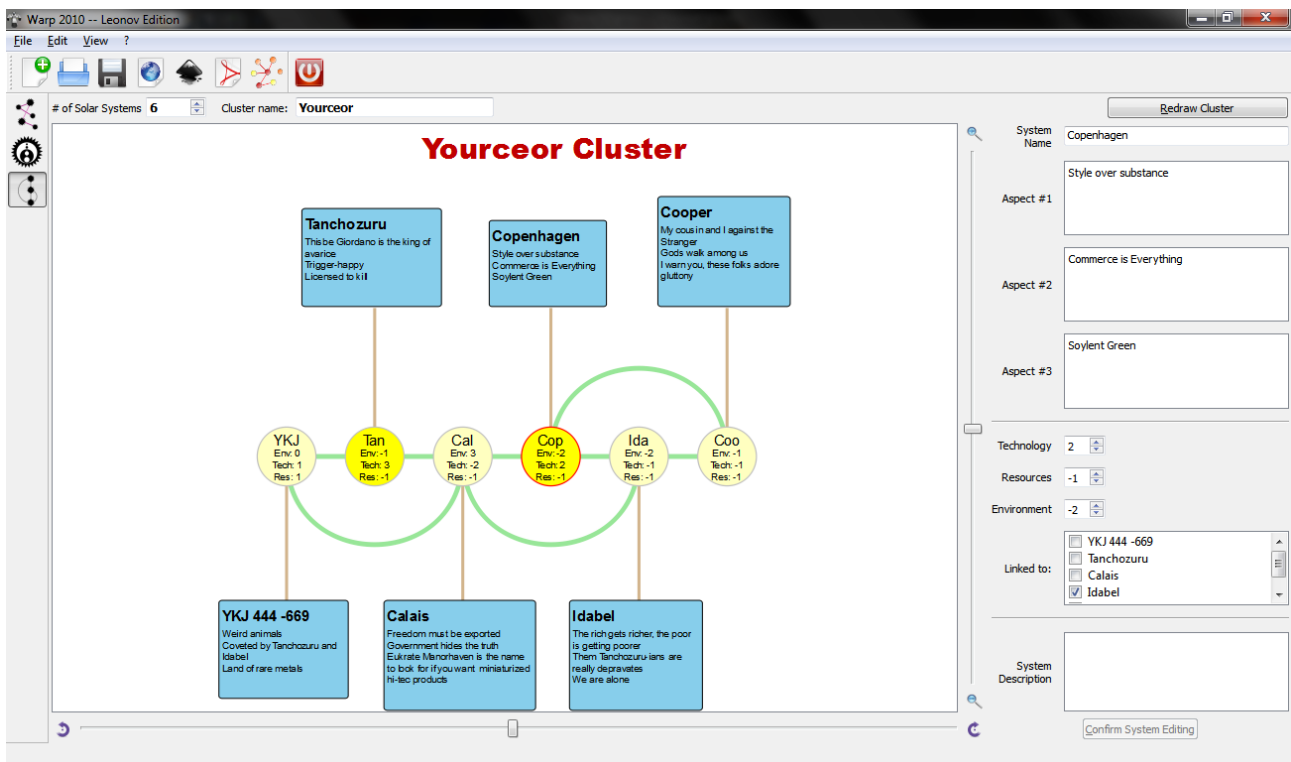
You create a brand new cluster by clicking on  icon. By default, you create a 6-system cluster, but you can increment the number of systems to create up to 18 by changing the value found on “# of solar system” input box.

Names are random-drawn, and are meant to be suggestions, as well as aspects. Don't worry about them sounding ridiculous or inappropriate: you can always change them.



## EDITING CLUSTER ITEMS


You can edit a single system by double-clicking on it:



The selected system then gets surrounded by a red line, and input boxes found on the right are filled by its data.





Just editing system values does not update system data: you have to commit your edits by clicking on “confirm system editing” button.

## SAVING CLUSTER DATA

Just click on  icon.

## PRINTING AND EXPORTING CLUSTER AS IMAGE

You can export Cluster data in a number of formats:

-  SVG – the cluster is exported in an SVG vector file.
-  PDF – as above, only you get a PDF document (neat, eh?)
-  GraphViz .dot file – for you nerdy, computer-toting graph-loving types.
-  HTML file – useful for RPG authors: cluster data is laid out in a tabular form, along with a SVG image of the cluster.

# ***Common Tasks***

## **PREFERENCES**

## **PAGE SETUP**

## ***Acknowledgements***