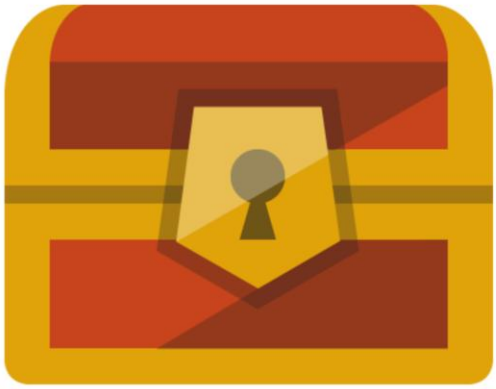
Sojour



By Rob Pollard

Contents

Preface

This manual includes screenshots from a made up RPG called **Window Guardians** to prevent any Copyright or IP infringements.

However, Sojour© can be used with *any* RPG system – it is rules light and can capture maps and create map tokens from almost any digital source thanks to Sojour's built in screen-shot functionality.

A number of assets were used in the production of this manual. Their sources are attributed below:

Character Portraits, Aiko and Arla: By **Dave Thaumavore** as part of **Dave's RPG Character Portraits (Fantasy 1: Humans)** which is available at DriveThruRPG.

These are published under the Creative Commons License (CC BY-SA).

Aias, **Cora**, **Evendur** and **Garion** are members of the player's party and use image numbers 04, 90, 08 and 43 respectively. These four images were modified by placing them in circular token frames.

Aiko is used to demonstrate token creation by using the screenshot tool. Her character uses image number 57. Her image was modified by reducing the size of the border to make her appear as a more natural fit in the dummy scenario book called **Kyle Manor**.

Arla is used to demonstrate transparencies. She uses image number 47. Her image was modified by putting it in a circular patterned frame.

Monster Tokens: These are part of **ProFantasy Software Ltd**'s Token Treasury: Monsters collection. The source tokens are Copyright **ProFantasy Software Ltd** and permission has been gained to include screenshots of them in this manual.

Maps: These were created by the author using **ProFantasy Software Ltd**'s excellent Campaign Cartographer 3+ software. There is no copyright on the maps. As an aside, this is the first time I have used Campaign Cartographer 3+, so don't judge it by the quality of my maps!

Window Guardians, North Sojourny Calendar, Window Guardian Character Sheets & Kyle Manor Scenario Book: These are all made up by the author to help illustrate various concepts in Sojour.

Sojour's EULA and this manual are located in this folder:

C:\Users\<< YOUR NAME>>\Documents\PollySoft\Sojour\Documents

Sojour uses Microsoft's WebView2 component to allow PDFs to be viewed. WebView2 uses some components under the BSD License. The Microsoft license and the BSD Notice can be found in Sojour's installation directory's WebView2 folder. This is located at:

<< INSTALLATION DRIVE>>:<<INSTALLATION FOLDER>>\WebView2

Introduction

Welcome to Sojour©, the premier solo role playing journaling software!

This is an indie solo project written by Rob Pollard, the author of this manual.

Before we continue, I should probably mention that I'm not a professional technical writer. The prose in this manual might not be up to the highest standards. For that I apologise. If you spot any obvious errors, whether technical or grammatical, feel free to contact me at the email address further down on this page.

Now back to the introduction…

I wrote Sojour because I found that the vast majority of existing Virtual Table Tops (VTTs) weren't really suited for solo play. They were also rather expensive, with some of them requiring one to re-buy content that one already owns!

Sojour is designed to aid solo role players with their gaming sessions by providing a means of journaling their adventures. It also keeps track of time and provides easy to use tooling to create maps and tokens directly from your digital scenario books (or anything else for that matter).

Sojour is system neutral and should be able to support any Role Playing Game (RPG).

The original design intent of Sojour is for the user to use their physical books and dice in conjunction with the software. That said, dice rollers, character sheets and other digital documents are supported, so you can play fully digitally, though this was never the original design intent.

I hope this software provides you with as much fun as it has me. If you have any ideas for enhancements that you would like to see, or perhaps some feedback, positive or negative, then please email me at sojour.pollysoft@outlook.com.

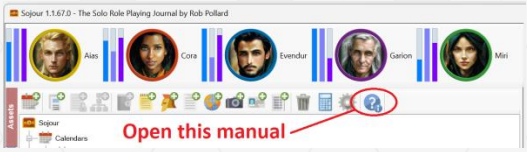
I can't promise that I can react immediately - as I'm only one person and this isn't my day-job. But I will endeavour to respond to all emails and will add all new feature requests to my management system for potential future inclusion.

With regard to DRM, I have made a conscious decision not to sell this software with DRM as I think it hurts legitimate buyers more than the pirates. So please respect this decision and do not make copies for others to use.

Finally, I'd like to thank you for purchasing this software, it means a lot and if enough people purchase it I can turn this into a full-time job. This will allow me to spend much more time and resources on it.

Getting Help

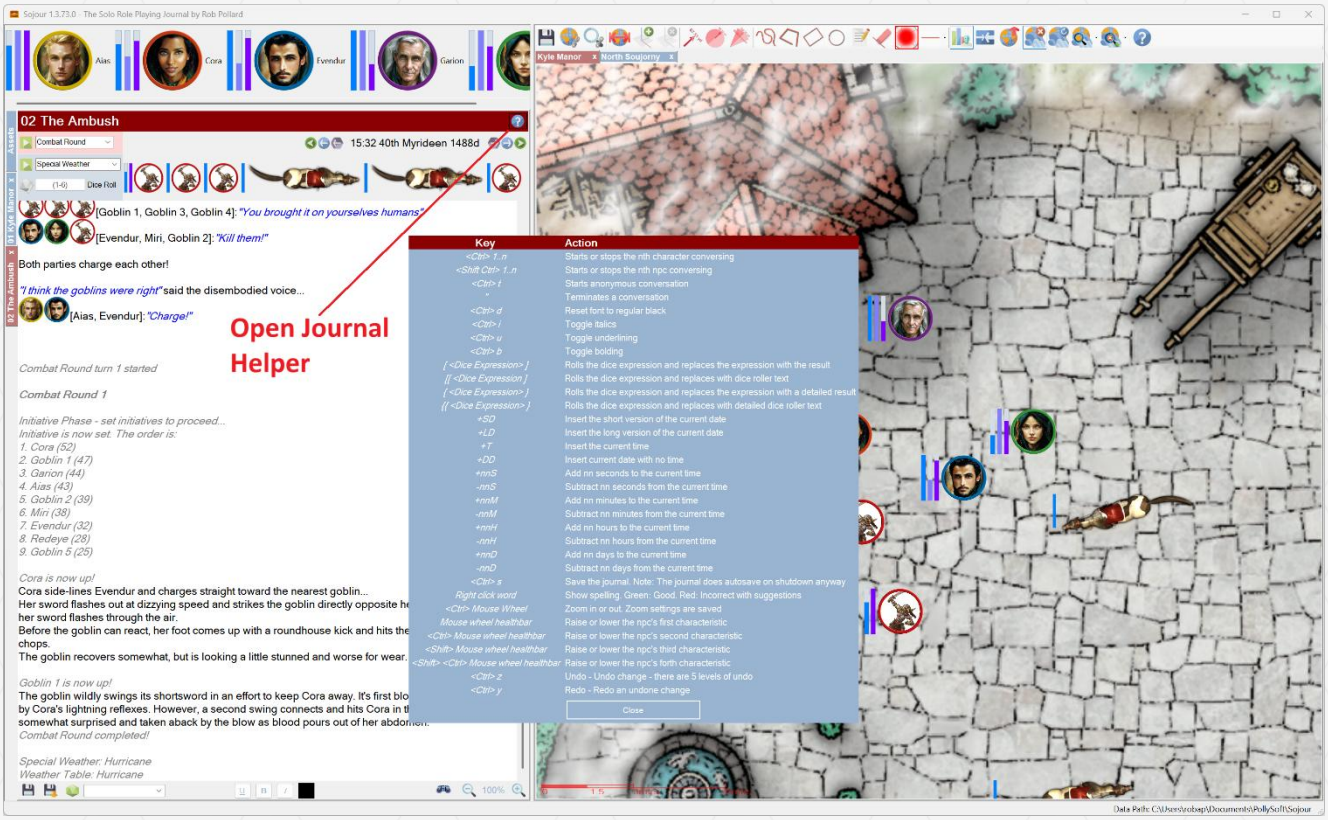
Sojour provides three general help resources. The first is this manual which is always accessible by clicking this button:



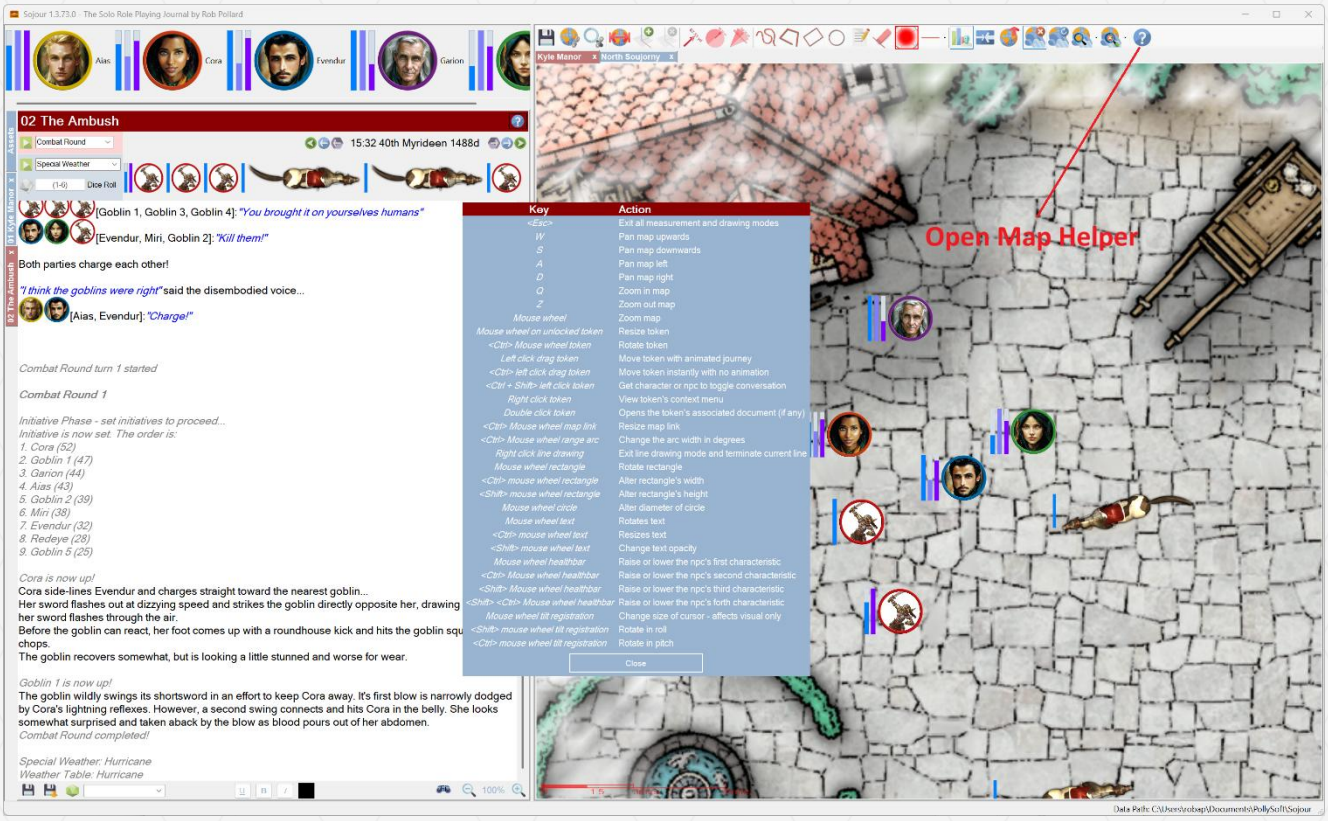
In addition to the manual, Sojour provides two keyboard shortcut panes, one for journals and one for maps. These are stay-on-top helper windows that provide useful information about how to manipulate those panes.

Both helper windows can be dragged anywhere and the system will remember where you last left them if you choose to reopen them again. These windows are also modeless, so you can continue to use Sojour when one, or both of them are open.

This is the helper window for journals:



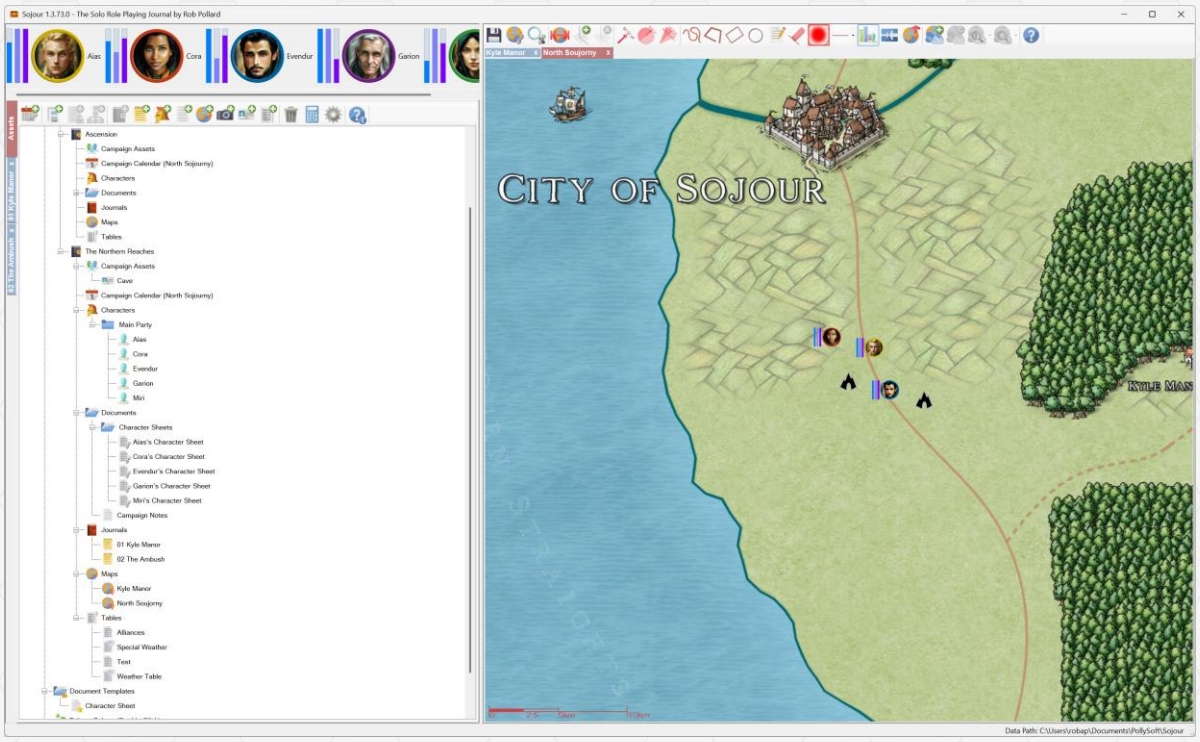
And this is the helper window for maps:



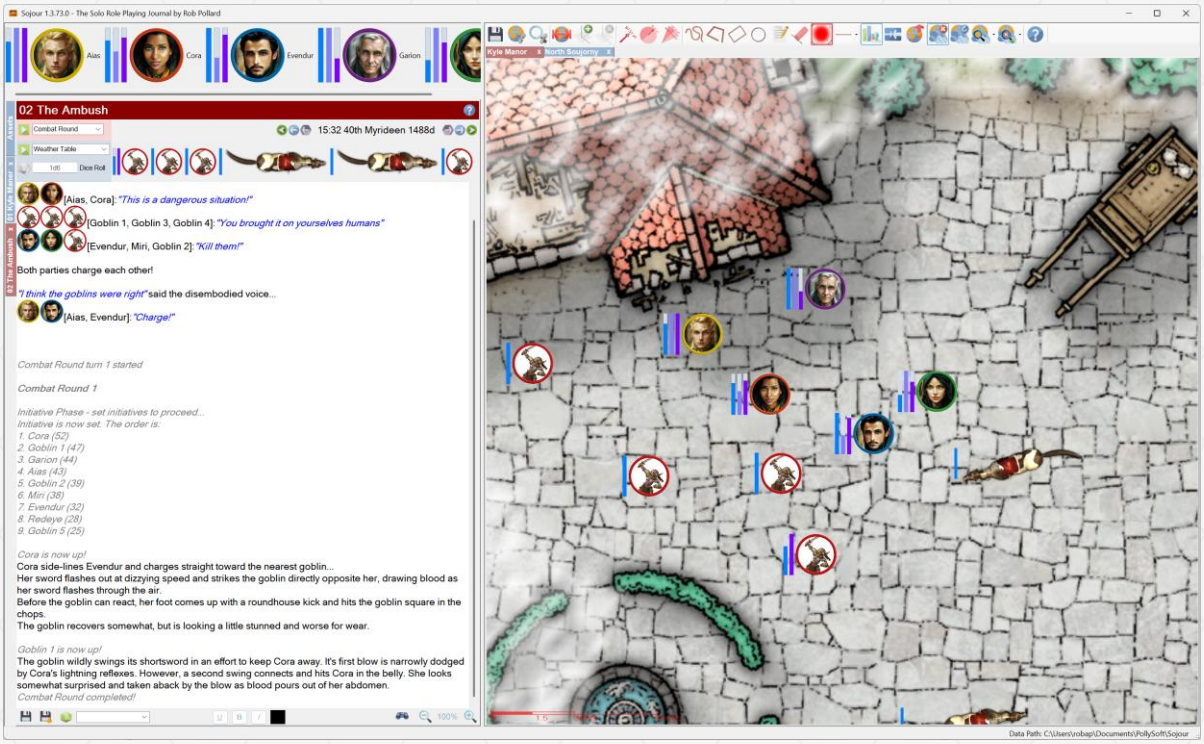
In addition to the built in help, I will be posting tutorial videos up on my You-Tube Channel. There is also a blog at Sojour – Sojour is the premier solo virtual table top (VTT). If you still have outstanding questions, feel free to email me at: sojour.pollysoft@outlook.com.

Sojour overview

Sojour's main screen is vertically divided into two halves.



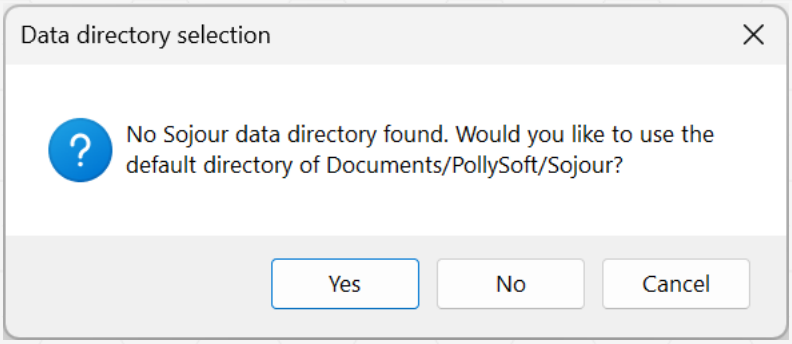
On the left is the **Assets Browser** pane; whilst on the right are the maps. In normal play, a journal will be opened up in the left hand pane, with an appropriate map displayed on the right, as shown below:



When you first start Sojour, it will appear quite barren. It will be up to you to add your own campaigns and adventures!

Starting Sojour for the first time

The first time you start Sojour you will be presented with this dialog window:

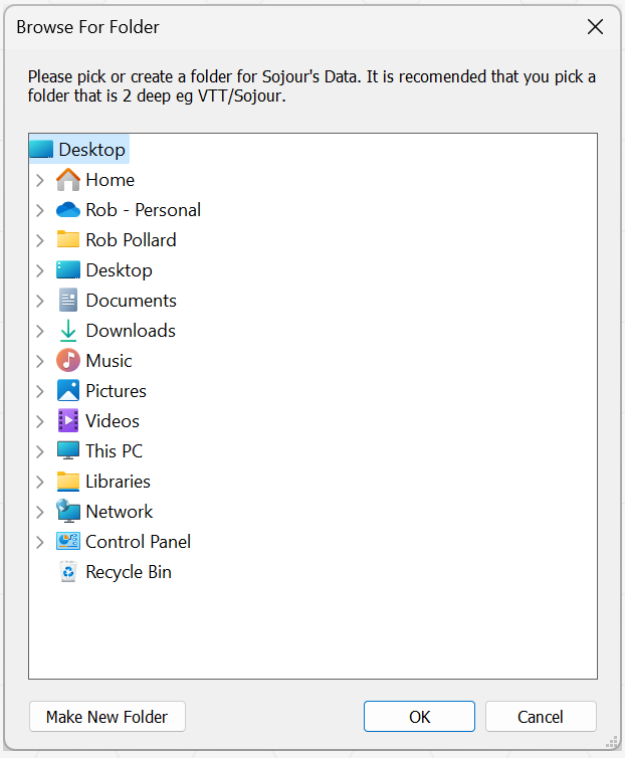


This window allows you to select where Sojour's data is saved. For most users, the default **Documents/Pollysoft/Sojour** is the best place as Microsoft recommends storing application data under that folder.

Click **Yes** if you wish to use this default directory. Sojour will then open up ready for you to use.

If you click **Cancel**, Sojour will close and exit.

If you select **No**, the following dialog window will be displayed:



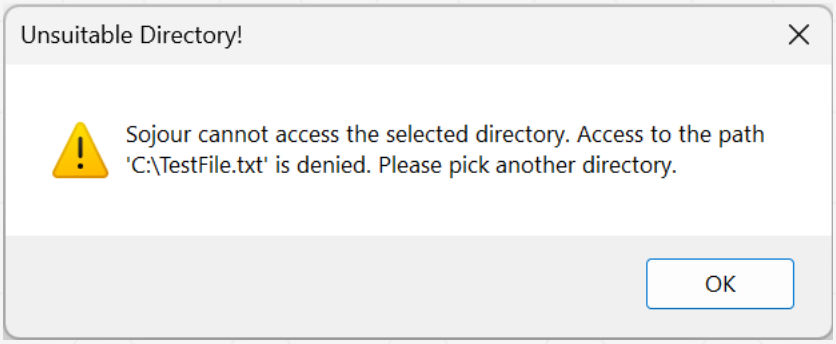
You can use this window to pick a directory or create directories that you can then subsequently pick.

The reason why the window recommends picking a folder structure that is two folders deep is that Sojour's data will go in the lowest folder and Sojour's backups from upgrades will go in the folder above.

For example if you pick a directory structure of **d:/VTT/Sojour**, Sojour's data will go into the **Sojour** folder, whilst any auto-backups will go into the **VTT** folder. The backup system does this because it can't assume the Sojour directory is in a good state (it's an extra safety feature).

For more information on these backups see the Automatic Data Upgrades section.

It is not mandatory to pick a folder two folders deep and Sojour will work just fine with any folder. However, Sojour will perform a file write check on your folder to see if it can access your chosen folder. If it runs into an issue you will see a window similar to this one:



TestFile.txt is a temporary file Sojour creates to test if the chosen directory is suitable. All sorts of error messages can appear here. In this example, the problem is that Sojour doesn't have permissions to write to the root of the c drive.

If you see this dialog window, you will need to pick another more suitable directory. At this point Sojour will take you back to the initial **Data directory selection** dialog.

Once you pick a valid folder, Sojour will start as normal.

Your data

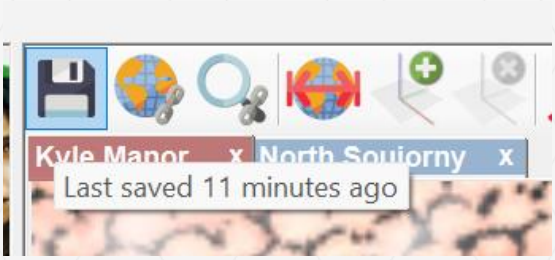
Sojour automatically saves everything except PDFs that you add or edit when you shut Sojour down. When restarted, it will attempt to restore itself just as you last left it. In addition to saving when it closes down, there are five other save modes to be aware of:

1. Sojour will save everything except PDF documents (see 4 below) when the main save button on the mapping toolbar is used. This is considered the primary means of saving your data.
2. All journals can be saved at any time using the journal save button, bottom left of the journal pane. This is purely optional, as the journal will be automatically saved when you close it or Sojour.
3. If you open the settings window, there is an auto-save option with a time interval defined in minutes. The system will attempt to auto-save in the background when this setting is enabled. This setting does not affect the normal auto-save on close – that always happens. It defaults to saving once an hour.
4. All PDF documents in the system have to be manually saved using the save button on their documents. This is a limitation of the 3rd party PDF software being used. See the documents design history section for more information.
5. Sojour will automatically perform a full save prior to importing any maps. This is because the import functionality is currently using some features of the Windows API and these features have a habit of crashing with very large image files (I am looking to replace this with a third party component.

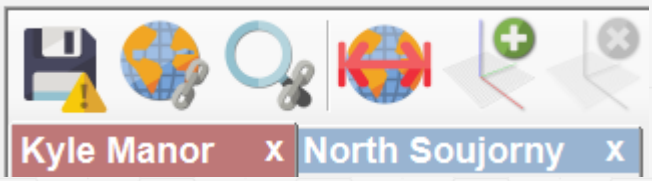
The Main Save Button

The main save button is a user's primary means of saving their data to disk. That said, Sojour will perform a full save when the application closes for in case a user forgets to save.

Hovering over the main save button will display a tooltip showing how long ago the last save was:

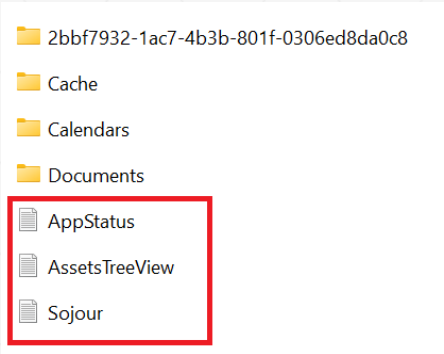


In addition, the main save button will display a warning if there have been significant changes made to Sojour that have not yet been saved:



The main save button is the only save button that saves the three critical data files used by Sojour:

1. **AppStatus.dat** – Stores which maps, tabs and documents are open.
2. **AssetsTreeView.dat** – Stores the node configuration of the assets browser.
3. **Sojour.dat** – Stores the whole structure of sojour including calendars, tables, rulesets and campaigns.



These three files are critical to the operation of Sojour. If any one of them gets damaged or corrupted Sojour can fail to load.

To that end Sojour has a loading and saving mechanism that is designed to protect the integrity of these three files. These two mechanisms are described below:

Saving

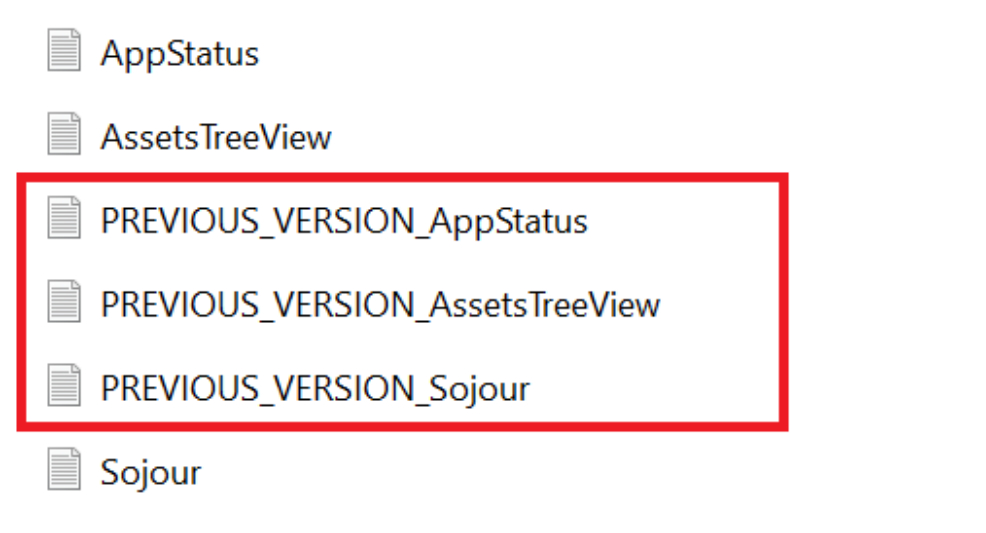
1. Sojour copies the target file to PREVIOUS_VERSION_<filename> if that file has been designated to be backed up (it's a Boolean on the save). This then becomes the backup.
2. Sojour then saves the in-memory data to a file called VERIFICATION_TEST_<filename>
3. Sojour then attempts to load VERIFICATION_TEST_<filename> to verify that it works.
4. If the verification fails, Sojour fails and throws. An error message will be displayed that the save failed.
5. If verification passes Sojour will copy VERIFICATION_TEST_<filename> over the original <filename> and VERIFICATION_TEST_<filename> is deleted.

Loading

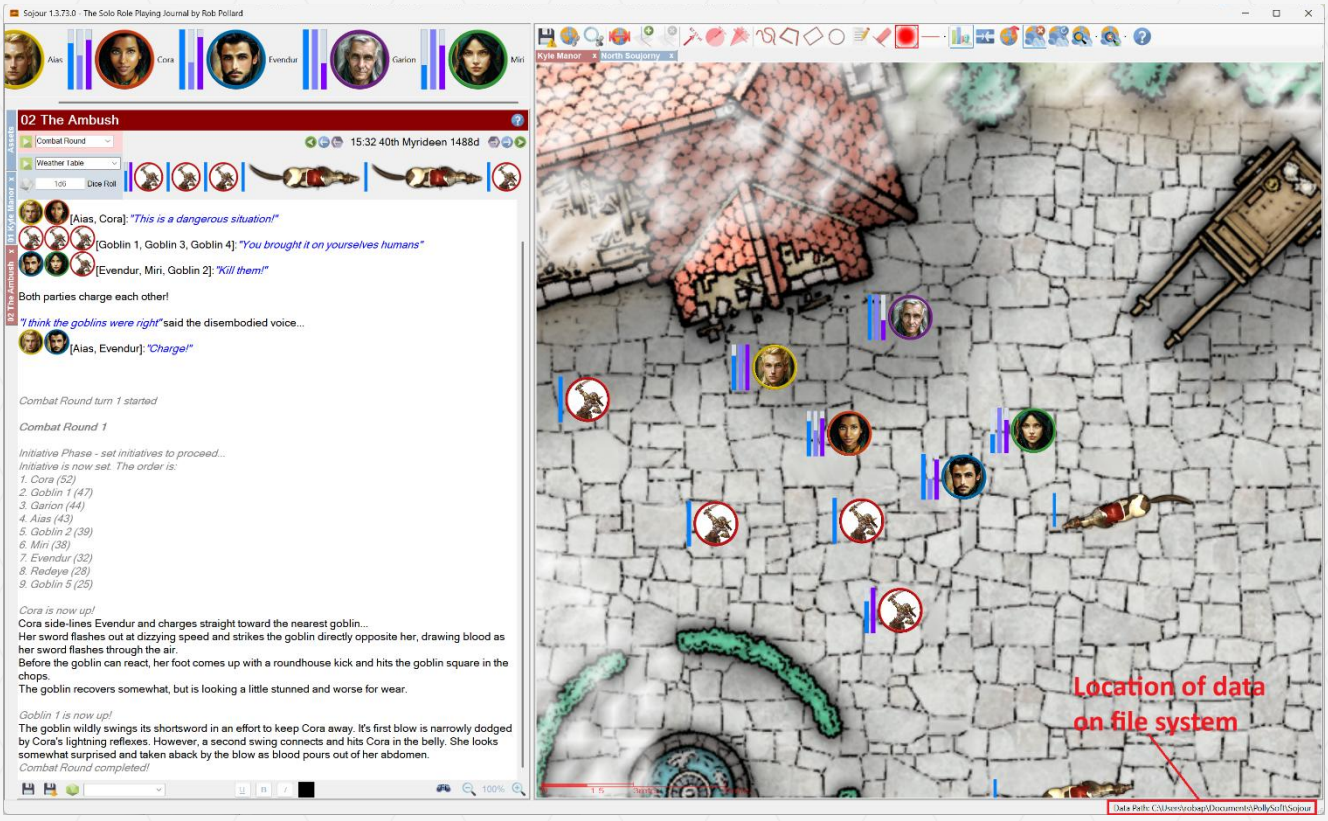
1. Sojour will attempt to load the target <filename>
2. If the load fails and Sojour does not detect a backup (PREVIOUS_VERSION), it will throw and take the user to the Loading Exception handler where they will be offered the choice to pick another directory or restore from backup.
3. If the load fails and Sojour does detect the backup, Sojour will display an error message and ask the user if they would like to load from their backup instead.
4. If the user says No, Sojour will throw and take the user to the Loading Exception handler where they will be offered the choice to pick another directory or restore from backup.
5. If the user says Yes:
    a. Sojour will delete any files with the name of ERRORED_<filename>.
    b. Sojour will attempt to load from PREVIOUS_VERSION_<filename> instead.
    c. If b. fails Sojour will throw and take the user to the Loading Exception handler where they will be offered the choice to pick another directory or restore from backup.
    d. If b. succeeds:
        i. Sojour will rename the existing and failed <filename> to ERRORED_<filename>
        ii. Sojour will then copy PREVIOUS_VERSION_<filename> to <filename> - so a copy of the backup becomes the normal save file.

The above mechanisms assure that Sojour's critical files are never overwritten, unless the overwriting data is valid.

It will also leave backup files behind prepended with PREVIOUS_VERSION_:



You can always tell where Sojour is saving its data by looking bottom right of the main window:

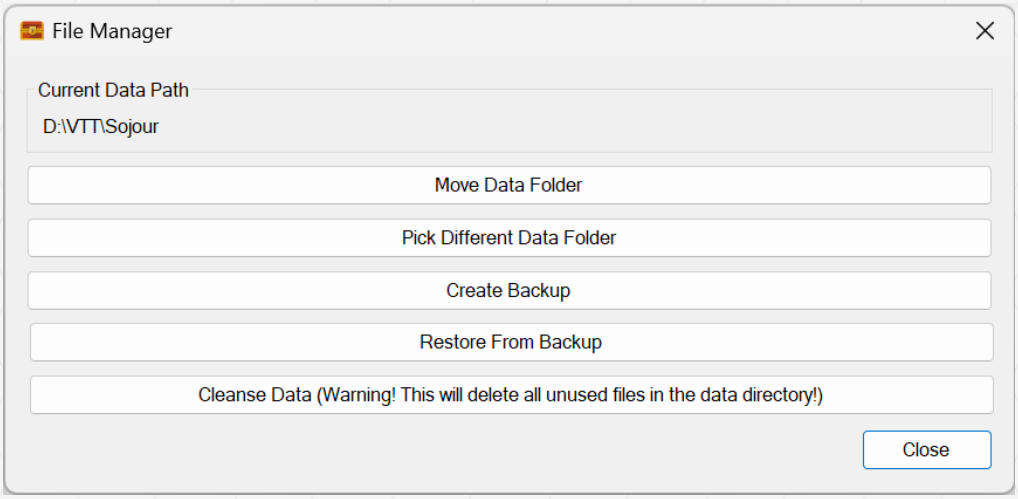


The File Manager

Sojour has a built in file manager to allows you to perform a number of operations with your data. You can access the file manager in one of three ways:

1. Double clicking the file system location in the lower right status bar of the main window.
2. Double clicking the **Sojour** top level node in the **Assets Browser**.
3. Right clicking the **Sojour** top level node in the **Assets Browser** and picking **Open File Manager…** from the context menu.

Once opened you will see this window:



The file manager allows five different operations to be carried out with your data, each of which is described below in its own sub section.

Move Data Folder

This option allows you to relocate your Sojour file data somewhere else. You will be presented with a folder browser to pick the relevant folder and drive location. Sojour will check it's a valid location and if it is, it will move your data to that new location.

If anything should go wrong, Sojour will revert back to your original folder location.

Pick Different Data Folder

This option allows the user to pick a completely different data folder. Picking this option will cause Sojour to save your current data. If the chosen folder already has data, that new data is loaded instead. If the chosen folder has no data, Sojour will start a new data instance in that folder.

This functionality is handy if you have multiple instances of Sojour data that you wish to switch between. Plus, as alluded to above, you can also use it to create new Sojour data instances.

Create Backup

This option takes a copy of your current Sojour data and copies it to your chosen location. It is recommended that the target folder is named in such a way as to remind you what kind of Sojour backup data resides there.

Backups are identical to normal Sojour data files and folders. So any Sojour data folder can also be used as a backup.

Restore From Backup

This option will **overwrite** your current Sojour data with backup data, so should only be used with caution. A warning dialog window will appear asking you if this is what you really want to do.

This is an ideal option if you use **Pick Different Data Folder** to create a new Sojour instance and then you want to populate that new instance with some Sojour data that you have backed up.

It can also be used to return the current saved data to an earlier state – assuming you have backed up that data first.

Cleanse Data

Sojour by default never deletes journal data from disk, even when you delete the journal in Sojour. This is because I consider your journals too valuable to delete.

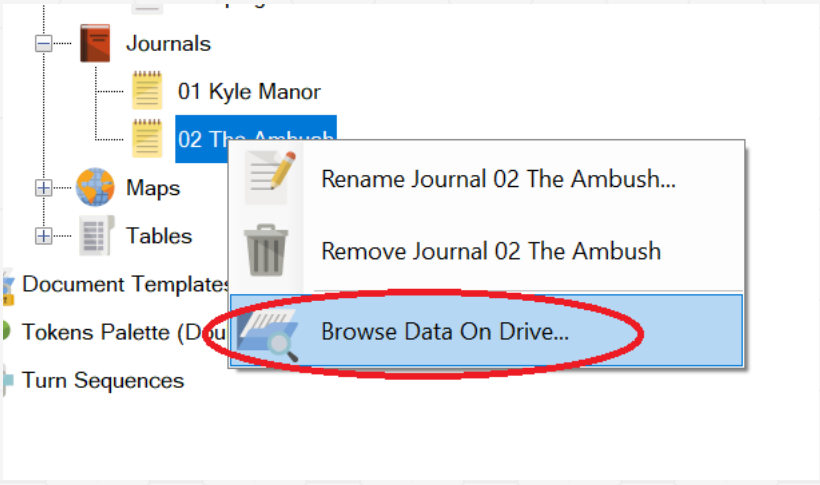
This option allows you to delete those old journal files. It will also delete all files and folders within Sojour's data directory and sub directories that are no longer being used by Sojour.

In addition it will run some cursory checking of your internal Sojour data's integrity and make alterations if it finds any issues.

Cleansing is not built into Sojour loading and saving as it will remove files that aren't being used, which might not be what you want.

Browsing your data

In addition, you can right click most items in the **Assets Browser** within Sojour to get access to the **Browse Data On Drive…** function which will take you straight to that item's data on your file system:



Sojour attempts to keep all files in a non-proprietary format to allow you to use that data in other software. After all, it is your data! However, there will be some assets where this is not possible, such as calendars. In these cases Sojour provides an import and export function.

Most Sojour specific assets can be exported and imported. However, you should be aware that the same code is used for all assets. To that end, it is worth naming your exported files to remind you what type of asset it is.

The reason for this, is that importing an asset under the wrong folder will not work, and may well lead to frustration. For example, trying to import a calendar under the turn sequences folder will fail!

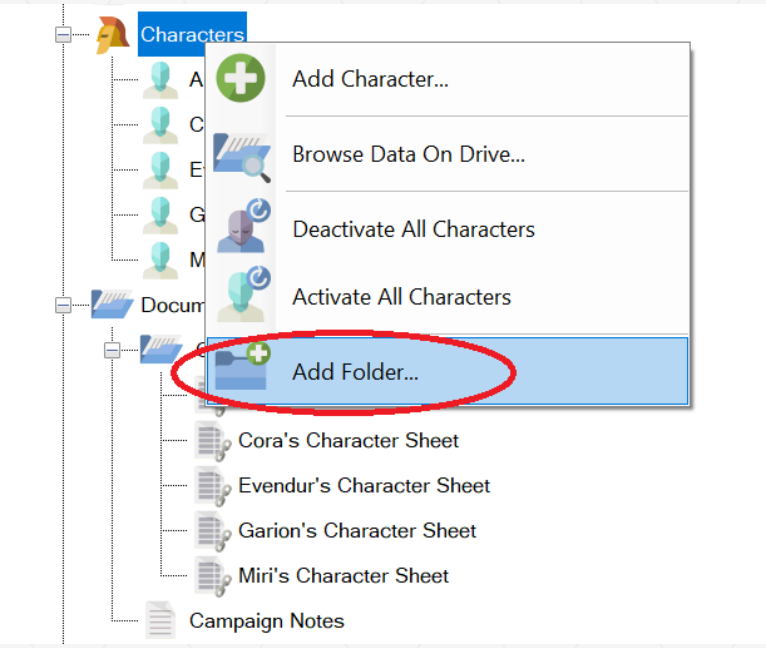
Naming your assets in such a way that the name reminds you of their type, should make picking the correct import folder easy.

Organising your data with custom folders

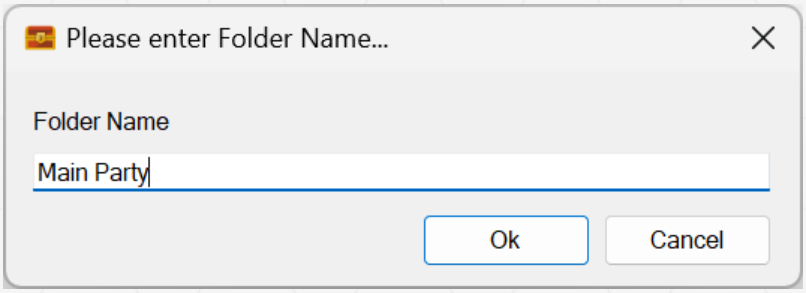
Whilst Sojour's folder structure is relatively rigid, it does allow the user some degree of organisation by offering the ability to create folders and sub-folders within the following **Assets Browser** nodes:

- Campaign Assets
- Characters
- Documents
- Journals
- Maps
- Tables
- Document Templates

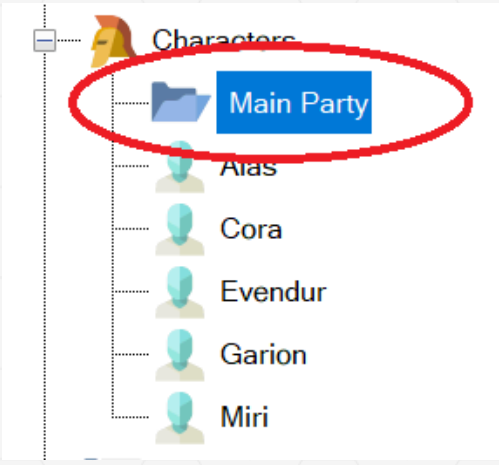
To create a folder, simply right click on one of the above nodes (or another folder) and select **Add Folder…**:



Selecting this option will display a dialog window where you can enter the folder name. In this example, we have chosen to add a folder to the **Characters** node and we are going to call it **Main Party**:

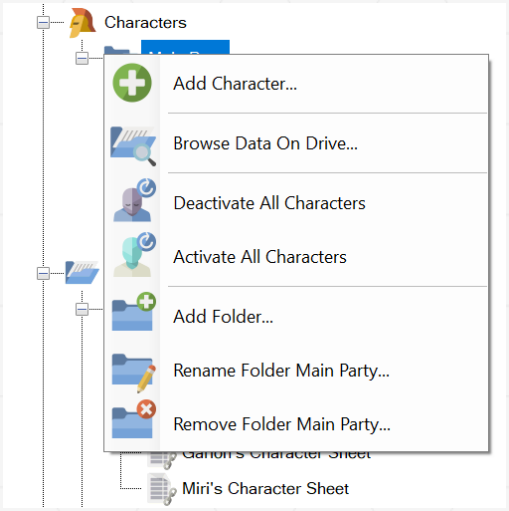


Clicking Ok will result in a new folder being created:



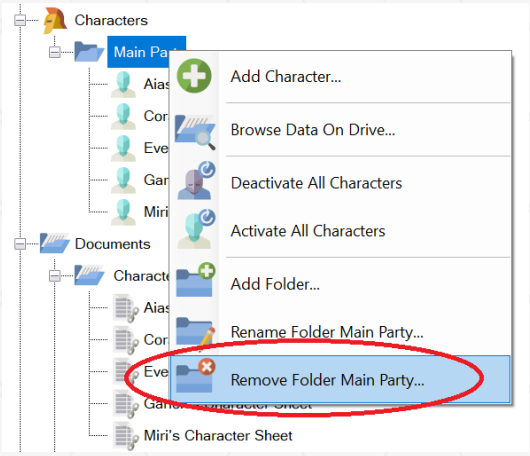
Folders and subfolders can only store the data type relevant to the top level node that they are under. For example, in this case, **Main Party** appears under the **Characters** node, so only characters can be stored in that folder and any other subfolders you choose to create under it.

Folders also inherit the menus from their top level node. For example, by right clicking the **Main Party** folder we can see a menu with all the same character based options as you would get by right clicking the **Characters** node itself:



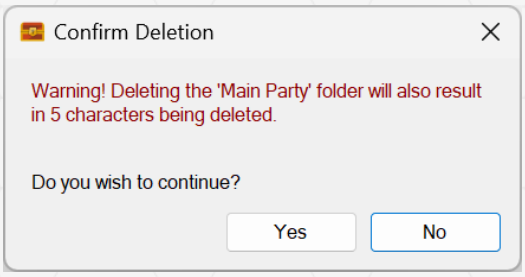
So in effect, anything you can do on the **Characters** node you can do on all its folders and sub-folders.

To remove a folder simply right click it and select **Remove Folder <Folder Name>**:



Removing a folder will remove that folder, all its subfolders and any assets held within.

If the selected folder does have any assets within it or its subfolders a warning will appear to allow you to back out from removing that folder:



In this example we are being warned that if we were to carry on with the removal of the **Main Party** folder we would also end up deleting the 5 characters under that folder.

The folders system also supports drag and drop. You have the ability to move folders by dragging them to other folders or nodes. However, there are a few limitations to this:

- You cannot drag a folder under another folder or node that is used for different asset type. For example, you could not drag a folder under the **Characters** node to a folder under the **Tables** node.
- You cannot drag folders between campaigns and rulesets.
- You cannot drag a parent folder into one of its own subfolders (this wouldn't make sense).
- You cannot drag a subfolder to its immediate parent (it's already there!)

You will know if a drag and drop operation is not possible because a 'No-Entry' sign will appear on the bottom right of the asset being dragged:

In the above example we cannot drag the **Main Party** folder to the **Characters** node because it is already there.

Dragging a folder will move that folder and all of its contents, including its sub-folders to the new destination.

Sojour provides no limitations on the number of folders, or their depth.

In addition to being able to drag and drop folders, you can also drag assets between folders and nodes. Just like folders, they also come with some drag and drop restrictions:

- You cannot drag an asset to a folder or node that does not support it. For example, you couldn't drag a character under a **Maps** node or its sub-folders.
- You cannot drag assets between campaigns and rulesets.

Campaign Assets and **Characters** can also be dragged directly onto the map.

Whilst these entities are being dragged over the **Assets Browser** they will appear as their original icon. However, once you drag them over a map, they will turn into an exactly scaled replica of the token that will be placed on the map if dropped:

In the above example we are dragging **Aias** to the map. Whilst over the **Assets Browser** he appears as a character icon with a no entry sign as he cannot be dropped here (shown on the left). However, once we drag him over the map – as shown on the right – that icon turns into Aias's token (and Vice-Versa).

Automatic Data Upgrades

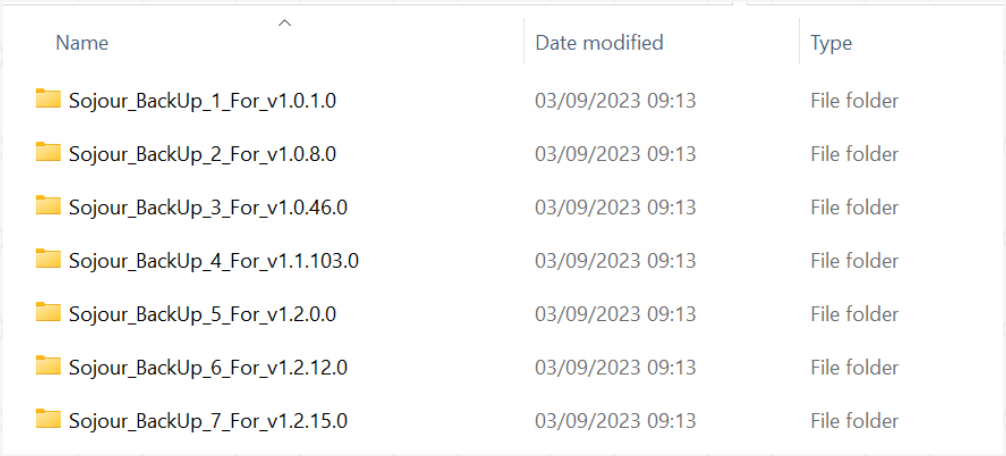
Occasionally, Sojour will need to update your data to make that data compatible with systems that are introduced or changed when upgrading Sojour.

Sojour always backs up your data *before* it makes these changes.

By default your backed up data is stored in:

C:\Users\<< YOUR NAME>>\Documents\PollySoft\Sojour Backups

Sojour will create one backup for each update applied. These backups are stored under the **Sojour Backups** folder:



These backups backup the entirety of your Sojour data folder and are simply the contents of the Sojour data folder but renamed. The content format is the same.

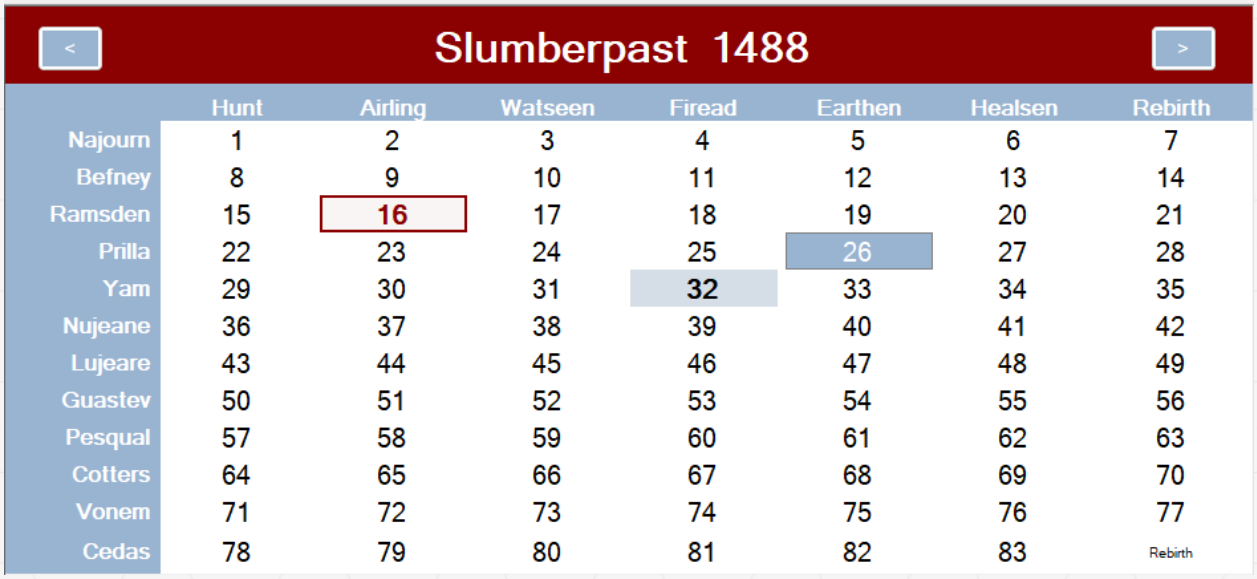
Each backup folder tells you which numbered update backed up the data as well as the latest Sojour version that supports that backed up data. For example, in the above screenshot, backup for update 2 is compatible with Sojour 1.0.8.0.

Before choosing to use a backup, you must have access to the installer that supports that backup. Follow the steps below to use a backup:

1. Uninstall Sojour and reinstall the version required to run the backup. If you don't do this, your backed-up data will simply be auto-upgraded to the version of Sojour you have installed.
2. Make sure Sojour is shut down.
3. Delete the Sojour directory under **C:\Users\<< YOUR NAME>>\Documents\PollySoft\.**
4. Copy your chosen backup to **C:\Users\<< YOUR NAME>>\Documents\PollySoft** then rename that backup's folder to **Sojour**.
5. Run up Sojour!

Given that Sojour stores most of its data in a non-proprietary format, you can always access many of your backup assets in other tools if needs be.

Calendars



Every RPG campaign is run against a calendar of some type. This is needed to allow for the tracking of time within the campaign or scenario.

Sojour has two types of calendar:

1. **Calendar definitions:** These appear under the **Calendars** folder of the **Assets Browser**. These define how each calendar works. We need these because many RPG's do not use the Gregorian calendar that is familiar to most Westerners on Earth. Note that a Gregorian Calendar is included by default, but it is not editable.
2. **Campaign Calendars:** These are an instance of a calendar definition. Each campaign will have just one calendar associated with it and this calendar is added to the campaign during creation. It is the campaign calendar that keeps track of the date and time within your campaigns. In addition, the system allows the user to create events which will appear in the journals when that particular day is reached.

For this manual, Calendar Definitions will be referred to as Calendars and Campaign Calendars will be referred to as Campaign Calendars.

Included Optional Calendars

Once you buy Sojour, a number of optional campaign calendars for a variety of commercial RPGs can be downloaded for use with Sojour.

Simply download the calendar you wish to use, then follow the instructions below:

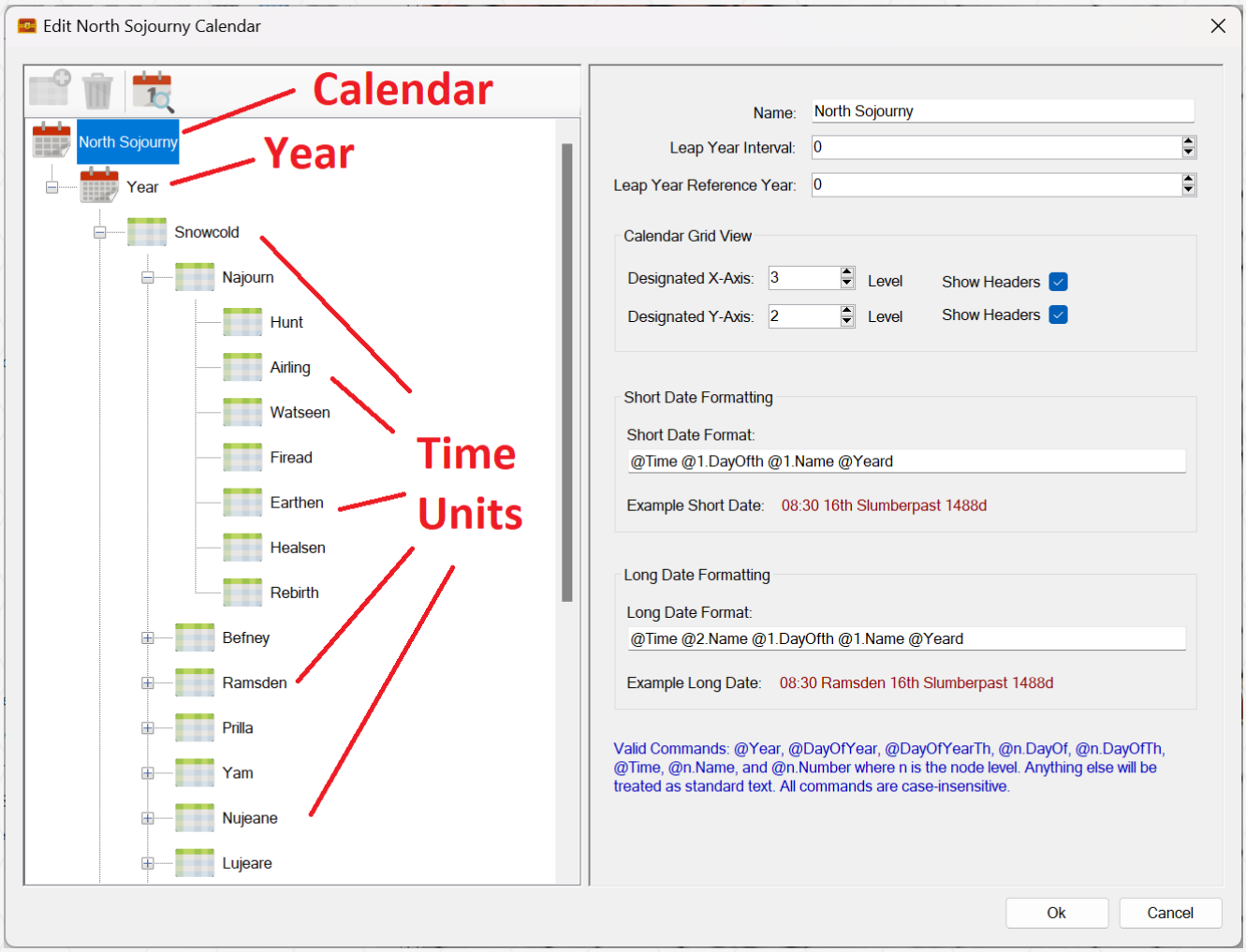
1. Right click on the **Calendars** node of the **Assets Browser** and select **Import Calendar…**. A file browser will appear that has opened up within Sojour's Data directory.
2. Navigate to the downloaded calendar and select it.
3. The newly imported calendar will appear under the **Calendars** folder ready for use!

Sojour Custom Calendars

Sojour does not limit you to these example calendars. It also provides tooling to allow you to create your own calendars. These tools can create pretty much any calendar, but there are some limitations you should be aware of:

1. Sojour can only create calendars that have the concept of a Year and of Days.
2. Sojour can only create time partitions, such as months, weeks, etc., that are made up of multiples of days – from 1 to n days.
3. Sojour cannot be used to create calendars where the day numbers are independent of the days. I.e. Each day must be associated with a specific day number. Note, this is not the case for the Gregorian calendar – but this is a special case calendar that cannot be deleted and is included with the system by default.

A Sojour calendar is composed of a calendar (called **Calendar**) at the top level, which in turn has a single year definition (called **Year**), which is in turn defined using one or more time units.



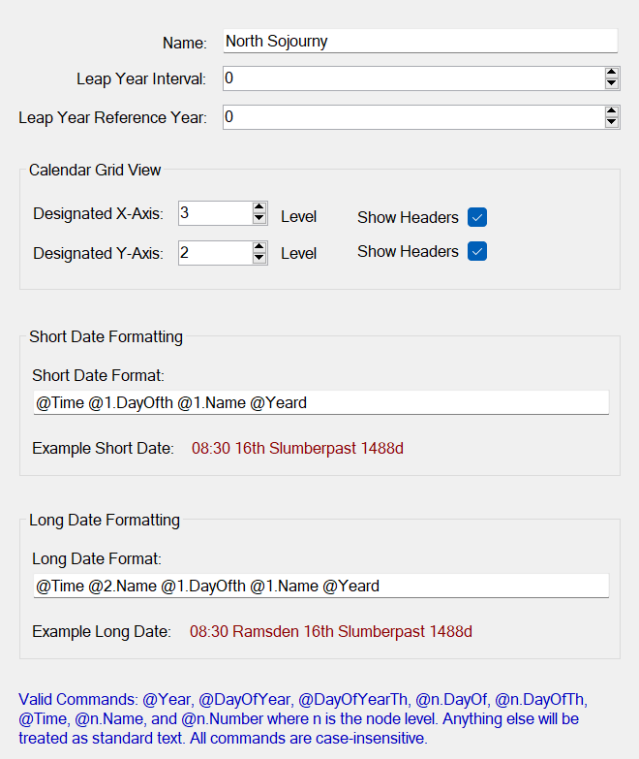
Each of these is described below.

Calendar

The calendar is the top level node and represents the calendar in its entirety. A calendar will always have one year definition within it that defines what a typical year for that calendar will look like. The Calendar node is typically renamed to the name of the calendar being represented. In the above case it is named 'North Sojourny'.

The calendar itself is defined with a number of fields which are described below.

Calendar Fields



Name: This defines the name of the calendar. The system will allow duplicate names to be used, and it will be able to tell them apart, but you might not be able to, so try and pick unique names for your calendars.

Leap Year Interval: This field defines how often a leap year comes around in years. It defaults to zero, which means there are no leap years for this calendar.

Leap Year Reference Year: This setting defines a reference leap year. Pick any year that is a leap year in your calendar. The system will then calculate all other leap years using this value with the **Leap Year Interval** value. For example if you set the reference to 1600 and the interval to 4, the system will auto-calculate 1604 as a leap year. This setting has no effect if the **Leap Year Interval** is zero.

Calendars have time units added to them which can represent any time unit smaller than a year. These time units can be marked as being visible only during a leap year.

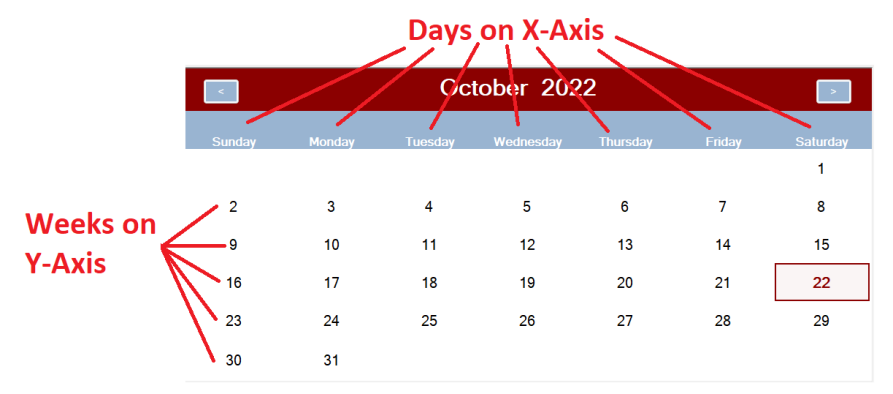
Calendar Grid View: This section defines how your calendar grid will appear when viewed. You can always get a preview of its appearance and functionality by clicking the **Example Date** button  on the toolbar.

This section contains two sets of fields, one for the X-Axis (horizontal) and the Y-Axis (vertical) of the calendar. These determine which level of time units are used to populate the X-Axis and which are used to populate the Y-Axis.

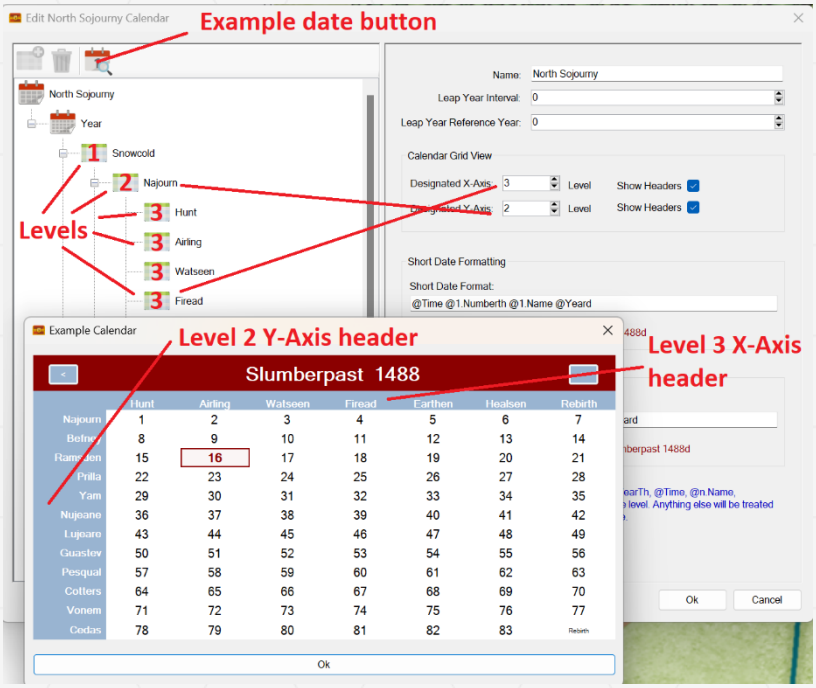
Warning! If you decide to set an axis level, both axis levels *must* be set, otherwise you will not see anything in the calendar preview. Displaying the light blue headers for either axis is completely optional.

You can think of time unit levels as differing units of time, with the lower numbered levels at the top representing larger time units. These time units can represent any time duration such as months, weeks and days – then it is up to you to decide which is used for which level. See the Time Units section for a further explanation.

For example, most real-life Gregorian calendars have days defined along the X-Axis and weeks on the Y-Axis.



If we take an example of the North Sojourny Calendar from Window Guardians RPG, we can see that this one uses both an X-Axis and a Y-Axis header as shown below:



Here the designer has set both headers to on. The Y-Axis is the vertical header, whilst the X-Axis is the horizontal header. All headers are shown in light blue and both are optional. They only appear if the relevant checkboxes are ticked.

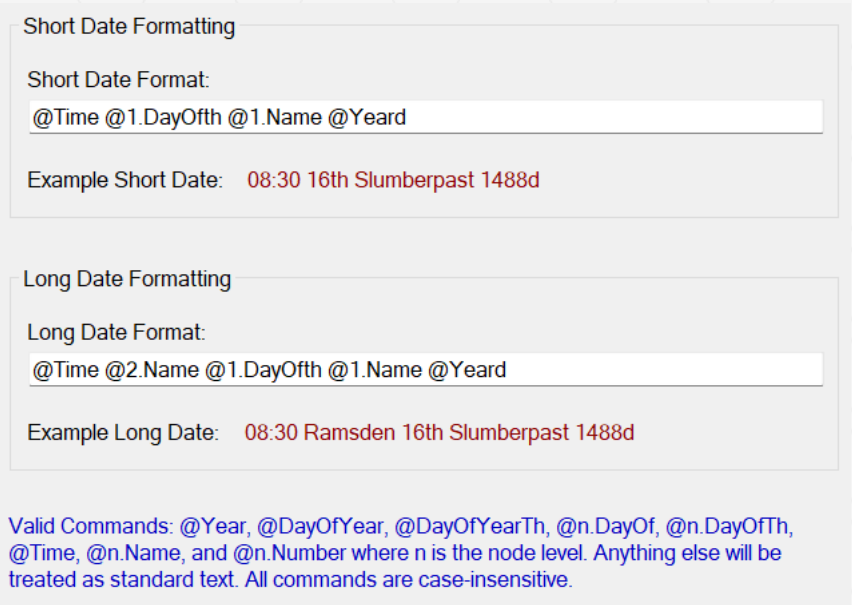
In the above example, the designer selected the level 3 time units for the X-Axis which are the days and the level 2 time units for the Y-Axis which are the weeks.

Feel free to experiment. You can't break anything (in theory) and you can preview your work by pressing the **Example Date** button shown in the screen shot above.

Short Date Formatting & Long Date Formatting: These two fields define how your calendar's date and times are expressed when written down as words and numbers. Sojour supports a short and long date format for each calendar.

Both fields are 'live' fields. As you type, you will see a real-time representation of your date in red below that field. This provides you with direct feedback as to how your dates will appear.

The fields accept all the commands shown at the bottom of the window in blue:



All commands are case-insensitive.

Any other text or number added to these fields that is not a command will be displayed as a literal. E.g., They will appear as written.

For example, the designer in the above screenshot added **d** to their date formats (after the year). As this is not a command, it is displayed as-is, as confirmed by the example red text.

The supported commands are as follows:

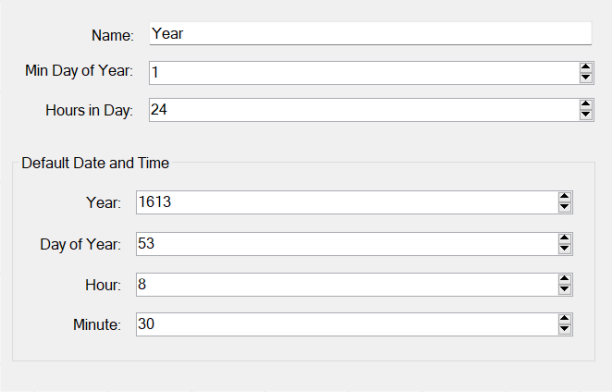
Command	Description
@Year	Displays the current year as a numeral
@DayOfYear	Displays the current day of the year as a numeral
@DayOfYearTh	Displays the current day of the year post-fixed with **st**, **nd, rd** or **th** as appropriate. For example 120 would appear as 120th and 121 would appear as 121st.
@n.DayOf	Displays the day of the selected time unit. So for a Gregorian like calendar it could be the day of the week, or the day of the month dependent on which level is selected.
@n.DayOfTh	The same as n.DayOf but post fixes the number with **st**, **nd, rd** or **th** as appropriate.
@Time	Displays the time in 24 hour format. For example: 21:30
@n.Name	Displays the current time unit's name. The n refers to the time unit's level in the hierarchy and must be a number between 1 and the depth of your hierarchy.
@n.Number	Displays the current time unit's number at the chosen hierarchical level. For example, if the first level of time units represents months and the current month was April, then selecting **@1.Number** would lead to a display of **4.**

Year

The **Year** node tells Sojour what a typical year looks like for the current calendar. A year will typically have one or more time units under it which define that year's composition.

In addition to the time units, a **Year** is defined by the fields shown below:

Year Fields



Name: Defaults to Year, but other calendars might refer to their years by a different name, so it can be changed here.

Min Day of the Year: This defaults to 1 as most years start on day 1. However, there are some calendars whose years start on different day numbers. A relatively common example, are the calendars that start their years at day 0. This setting allows you to change that start day. Note this setting only has an impact on the day of the year count.

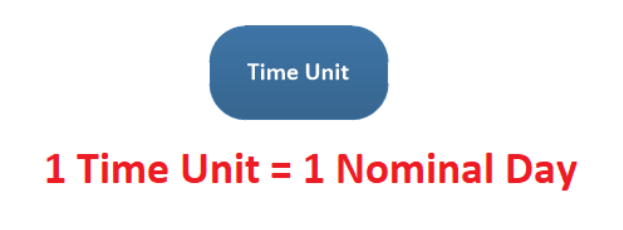
Hours in Day: Defines the number of hours in a standard day. Defaults to 24.

Default Date and Time: This section allows you to define the default date and time whenever a new campaign is created using this calendar. The dates and times can be changed later, but it can speed things along to provide a good ballpark default.

Time units

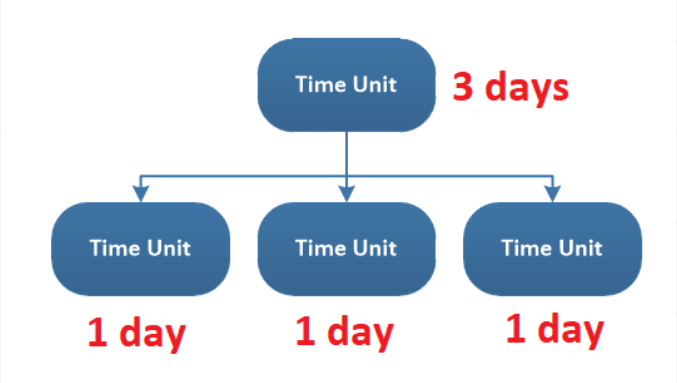
Every calendar is composed of one year definition, which in turn is defined using one or more time units. These time units can represent days, months, weeks or anything else you can think of.

Time units are generic blocks of time that are nominally one day in length.

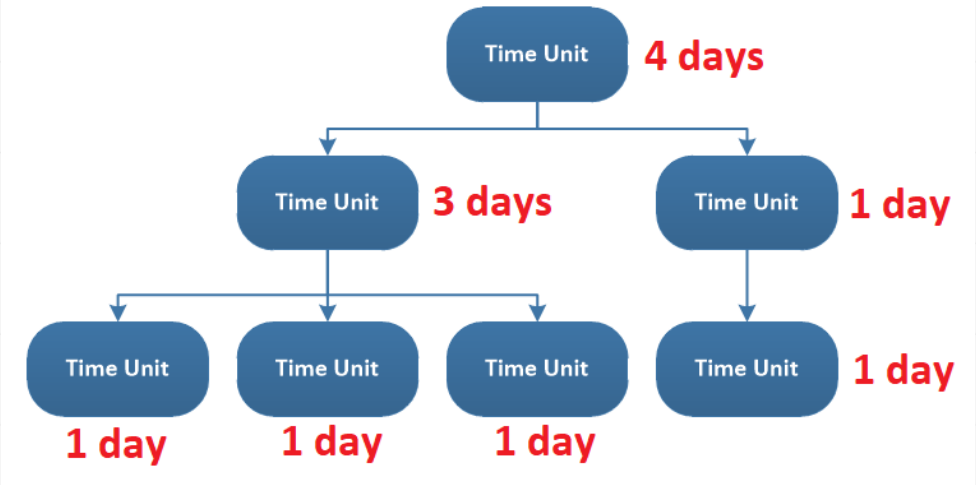


Time units can be placed inside (or under) other time units to become that time unit's children. The number of children defines the length of time of the parent.

For example if a parent time unit has three time units under it, the parent's duration would be 3 days:

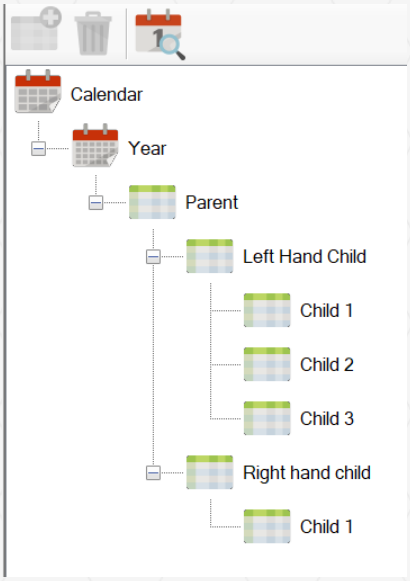


You are not limited to just the immediate children either. You can add time units to the children too! Consider the image below:



In this example the parent time unit (the one at the top) is 4 days in length and its first child on the left is 3 days in length, with the second child on the right being 1 day in length. Each of the lowest level time units always represent exactly one day.

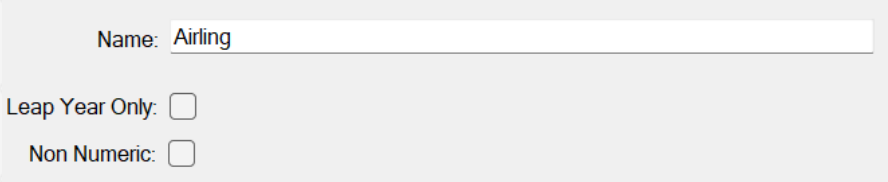
How would the above structure appear in a calendar within Sojour? See the screenshot below:



Each time unit can have a custom name, like Week, Month etc. This makes time units extremely flexible. They should be able to model almost any calendar!

Time units are composed of the fields below:

Time Unit Fields



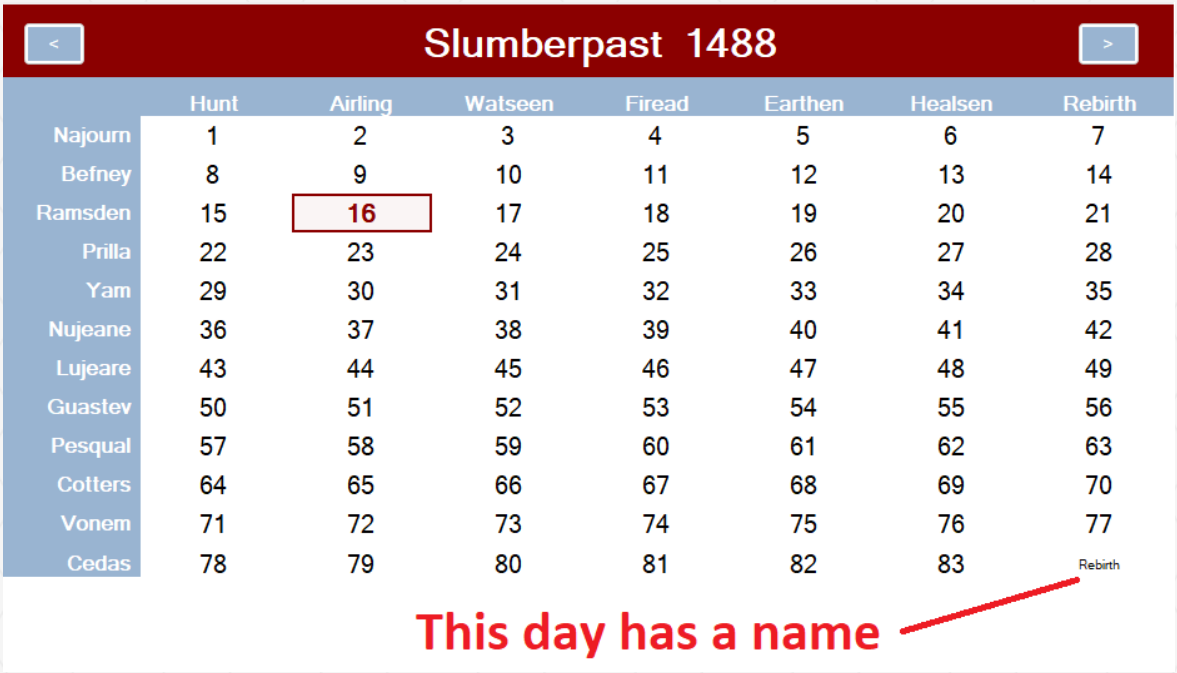
Name: The name of the time unit – typically a unit of time E.g. Month or Day or a time unit name E.g. June or Sunday.

Leap Year Only: Mark this time unit and all its children as only being available when it is a leap year.

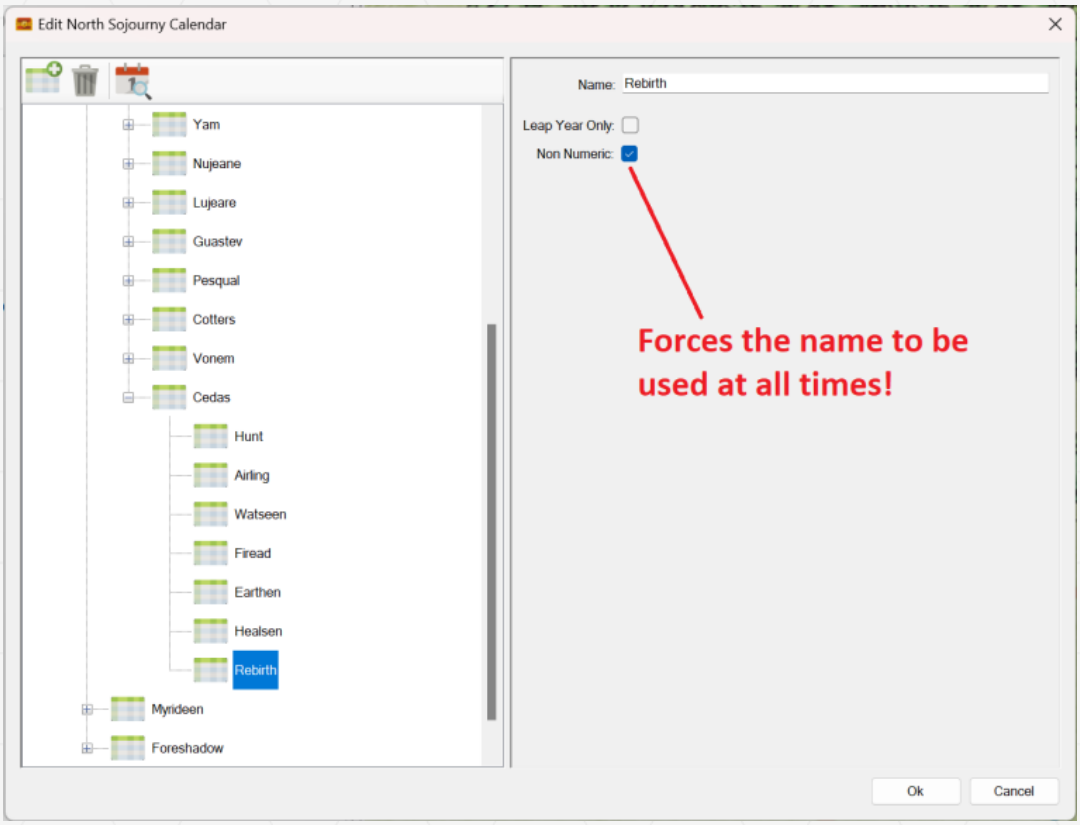
Warning! If you set **Leap Year Only** and do not set the year to have leap years, you will never see those time units in your calendar!

Non-Numeric (lowest level time units only): By default, time units that appear in headers, either vertical or horizontal on the grid view, use their names for display purposes; otherwise they are displayed as numbers. These numbers are counted from the current time unit back to the first one at its level. Ticking **Non-Numeric** allows you to override this behaviour and have the time unit always display its name.

For example, consider the calendar below, one of the days has a name, rather than a number:



By default all days that are not in a header (the blue area) are displayed as numerals. In the above case, **Rebirth** would normally be displayed as 84, but in this case we have ticked the non-numeric checkbox to force it to always use its name:

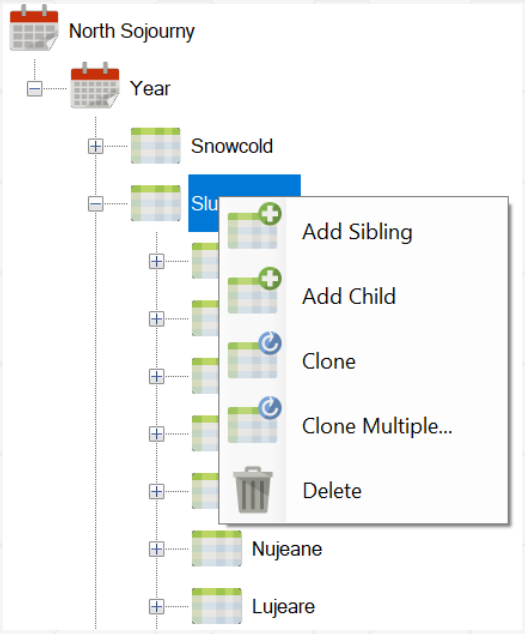


It is recommended that you browse through the three example calendars to get an idea how all these concepts fit together.

Creating a calendar using time units

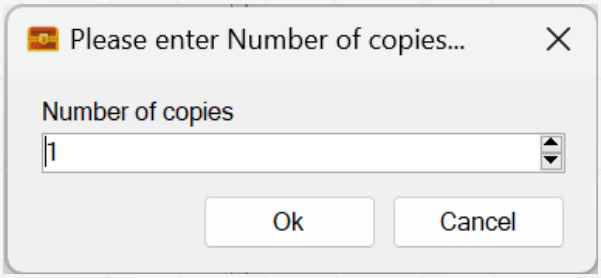
The hardest part to creating a new calendar is working out how to arrange your 1 day time units. A calendar must have one time unit for every day of the year. For example, if your year has 365 days in it, your calendar must have at least 365 time units.

Having so many time units in a calendar sounds like a laborious task, however, Sojour provides various clone and copy functions to make adding them a quick and efficient process.



Right clicking on a node allows you to add a child time unit, a sibling – one that is at the same level, or you can clone it once or clone it multiple times.

The clone functions are recursive, so if you clone a top level node, the clone will also clone all its children too, thus saving a lot of time. When you choose to clone you will see this dialog window:



Simply enter the number of *additional* copies of the node that you would like and hit enter. Sojour will automatically clone those nodes for you at the same node level as the original.

As mentioned previously, the best way to see how the calendars are put together is to open up one of the example calendars and explore what's there.

Rulesets

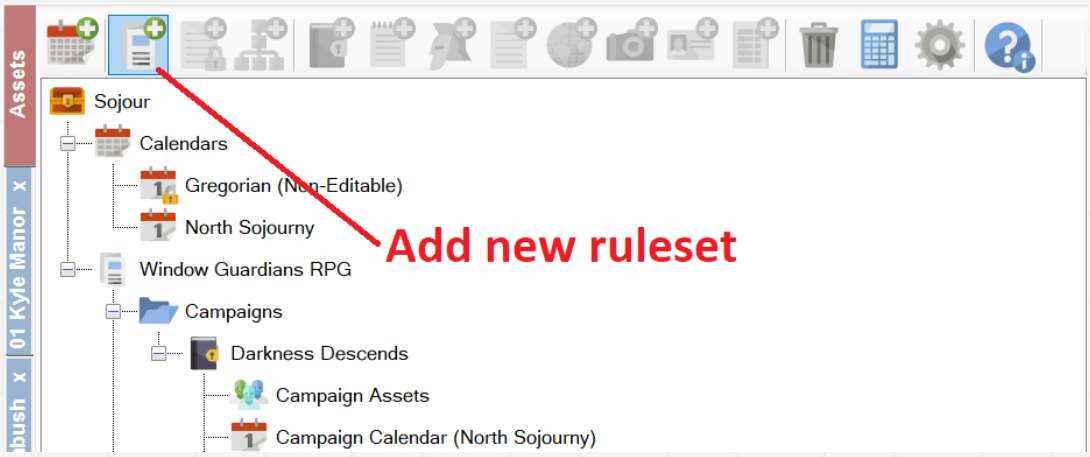
A ruleset represents a set of Role Playing Game (RPG) rules. Sojour is a rules-light system in that it provides very little automation for your rules.

Sojour was designed this way because the intent is that you would use your physical rules and dice or perhaps digital versions of them. Sojour would then provide you with the journals, maps and time tracking tools to use in conjunction with those rules and dice.

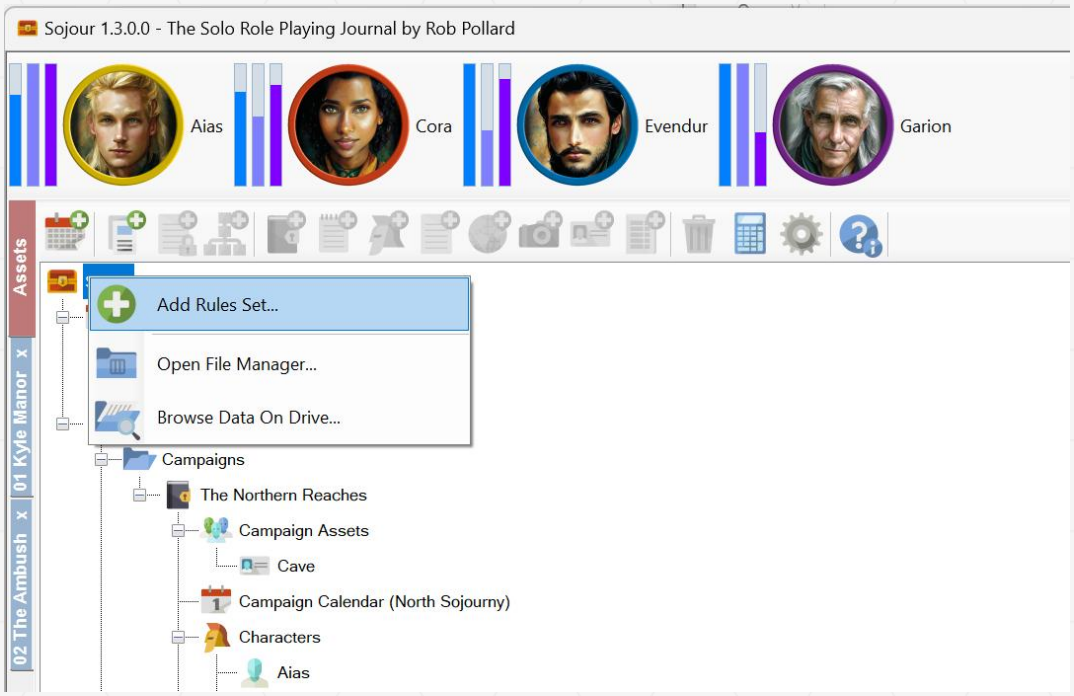
All campaigns, and ultimately adventures, must be part of a ruleset. The ruleset provides a container for these, helping you to keep your games organised.

Adding a new ruleset

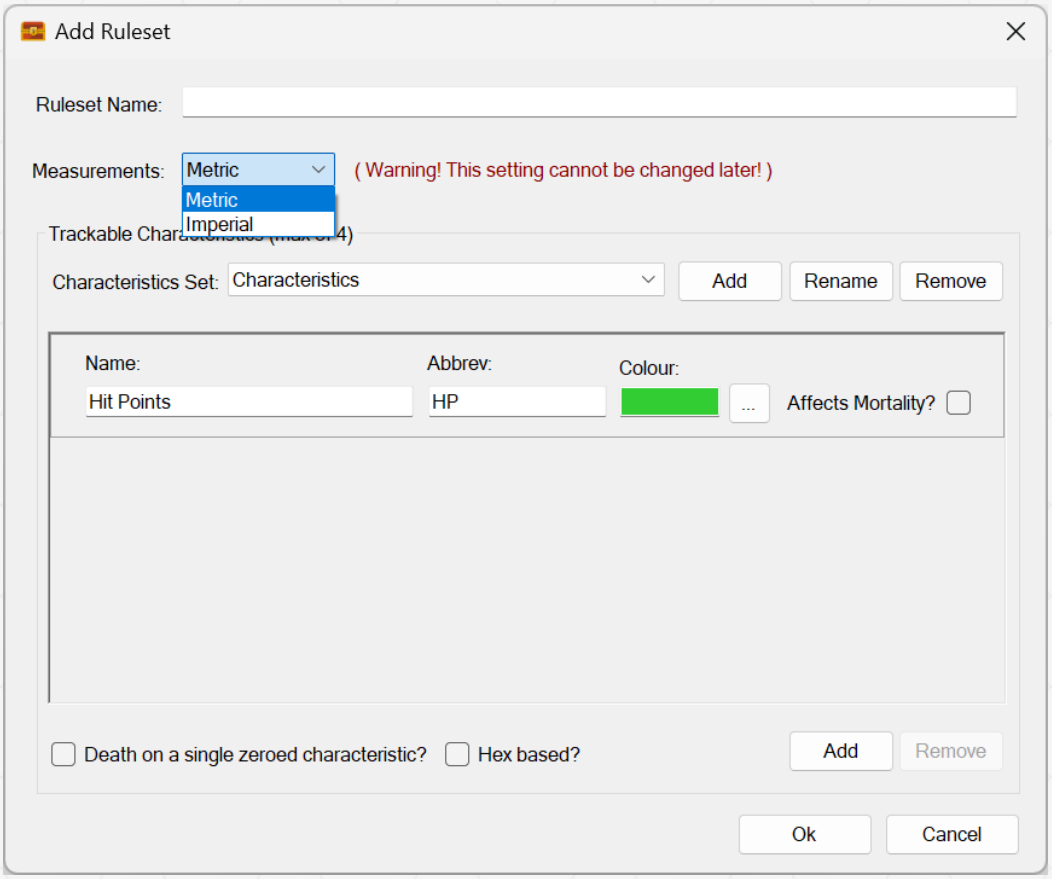
Creating and adding a new ruleset is easy. You can click the **Add Ruleset** button in the main menu:



Or alternatively you can right click on the **Sojour** node to pull up the context sensitive menu and add your ruleset from there:



Once you add a new ruleset, you will need to give it a name and pick its measurement system:



The measurement system is important in that all map measurements will be based on your chosen selection. At the time of writing, this cannot be changed after the fact, so be sure to check your rule books to make sure that the correct measurement system is chosen.

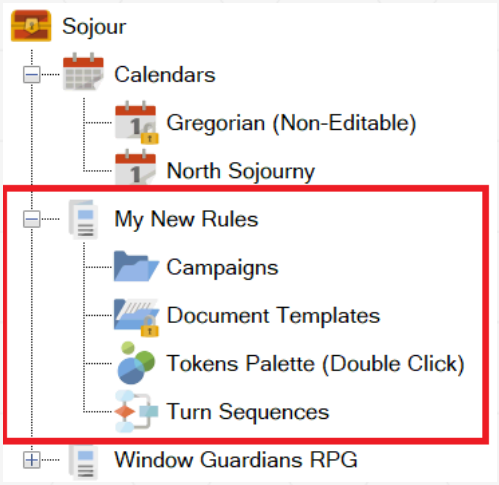
Below measurements are characteristic sets and trackable characteristics. These can also be setup at this time too, though these can also be edited later if required. Trackable characteristics represent changing values associated with characters, npcs or campaign assets that you wish to keep track of.

Sojour supports up to four trackable characteristics per set. These will be shown as health-bars on the portraits and on the map.

Characteristic sets allow you to create multiple sets of characteristics. For example, you might have a set for characters, another set for monsters and maybe an additional set for vehicles. It's up to you!

Trackable characteristics will be explained in more detail in their own section below.

A newly added ruleset will create four brand new folders under it:



1. **Campaigns**: This folder stores all the campaigns you are going to be running under that ruleset. All journals and maps reside within a campaign. More information can be found in the Campaigns and Journals sections.
2. **Document Templates**: This folder stores PDF or RTF document templates that are used with your chosen ruleset. A typical one that most rulesets have is a character sheet. More information can be found here.
3. **Tokens Palette**: This palette is used to store map tokens and their associated document for the ruleset. In general this is used for storing your mobs or monsters that need to be available in every campaign. More information can be found in the Tokens, characters and campaign assets section.
4. **Turn Sequences**: These are used to define what a typical turn looks like. For most RPG's this will normally be combat turns, but there are others. More information can be found in the Turn Sequences section.

Trackable Characteristics

Every rules set can have zero or more sets of trackable characteristics assigned to it. These are added when creating a ruleset. Trackable characteristics are used to provide custom health-bars for characters, npcs and campaign assets.

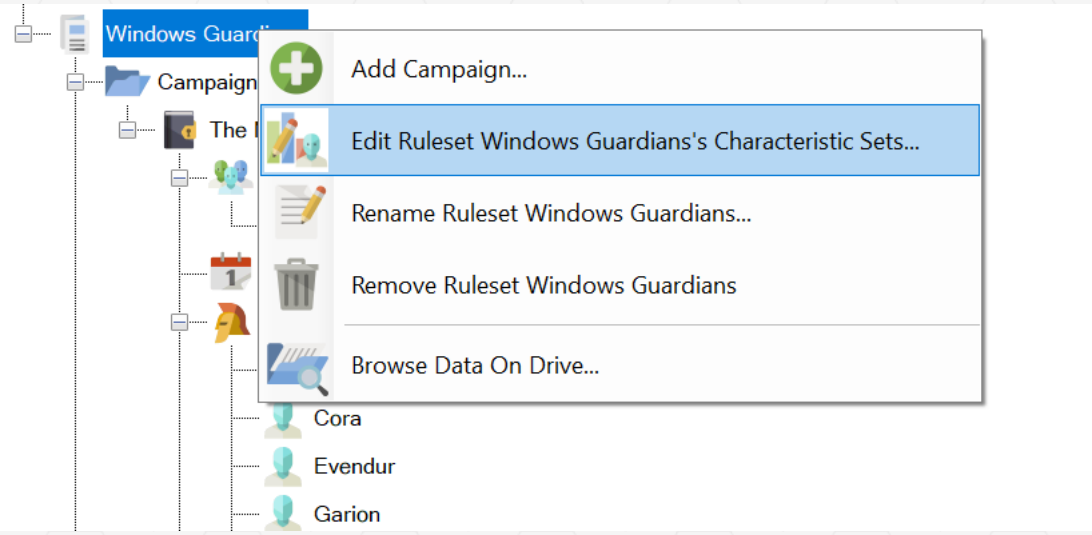
Up to four trackable characteristics can be defined for each set and only one set can be allocated to a character, npc or campaign asset.

The health-bars can then be manipulated with the mouse wheel to change their values.

Characteristic sets and trackable characteristics are also used to set up the criteria for defining death. When death occurs, the relevant map token can be removed from the map if that option is enabled in Sojour's general settings.

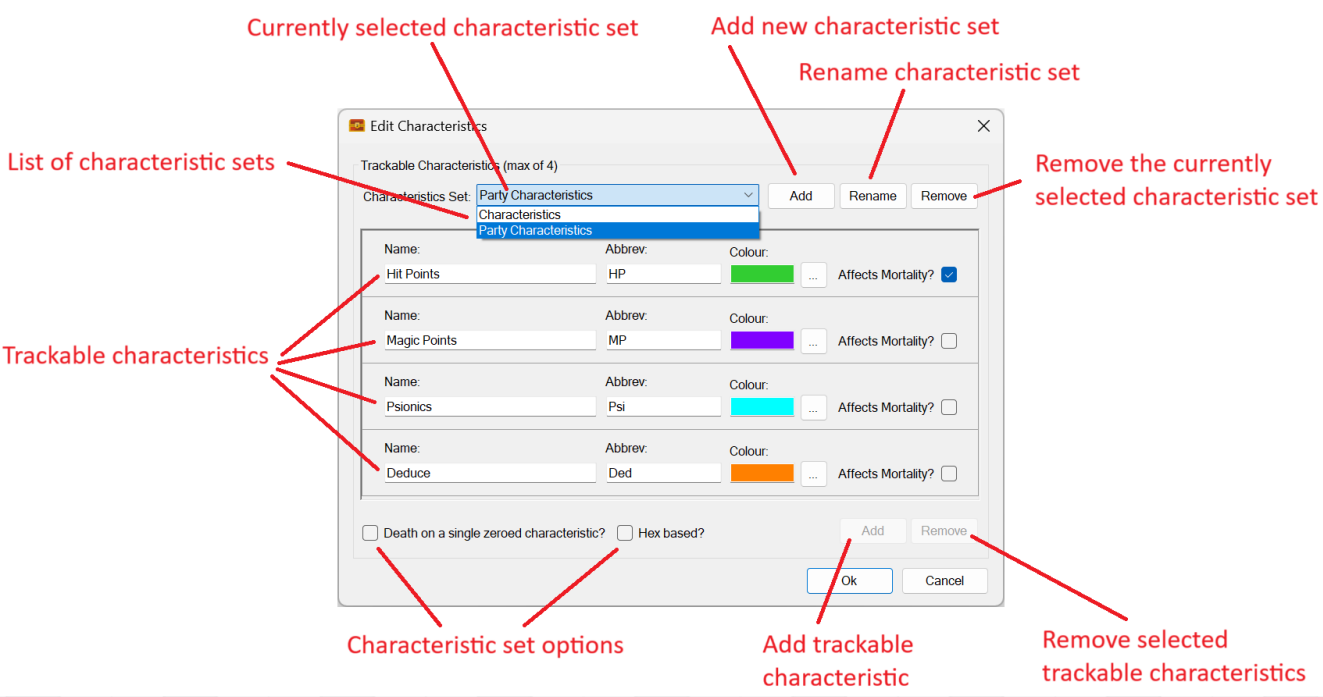
Characteristic sets are defined when a ruleset is created. However, they can also be edited after the ruleset is created from two addition places:

1. By right clicking the ruleset and selecting **Edit <Ruleset Name> Characteristic Sets… :**



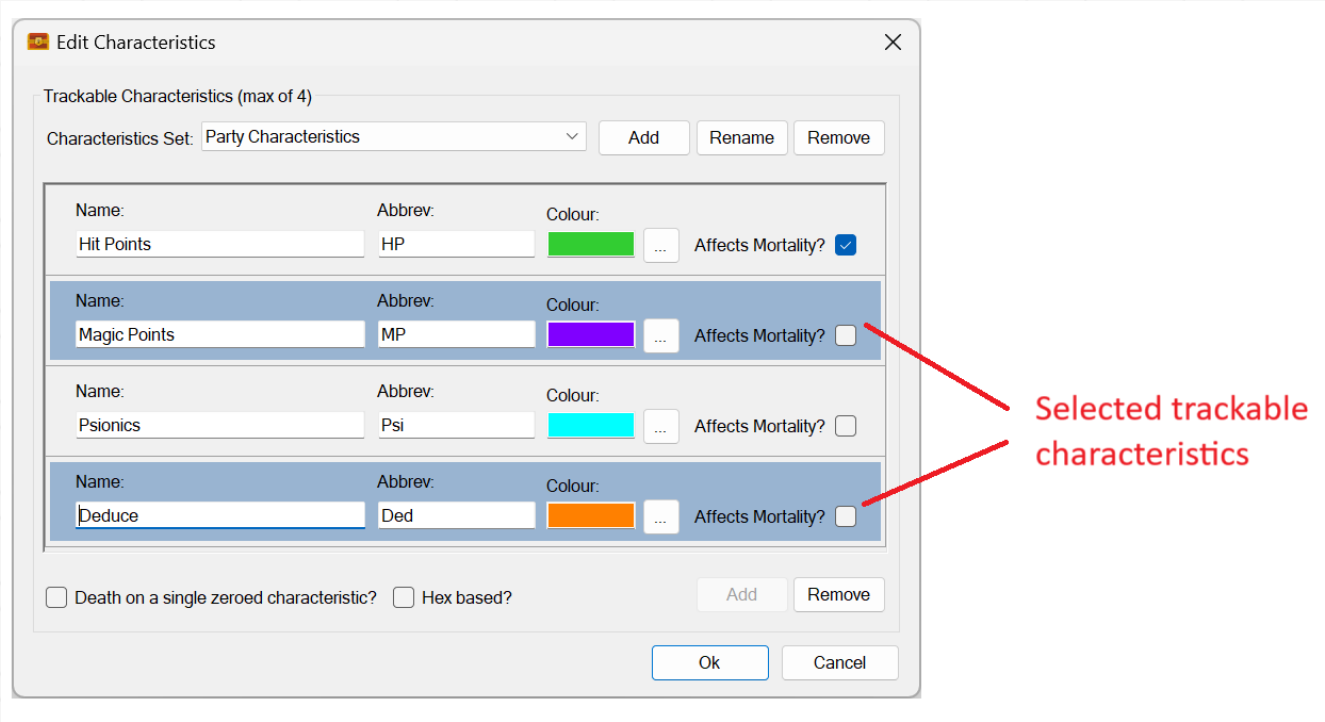
2. Or by opening a characters', npcs', campaign assets' or tokens' details page.

However you get there you will be presented with this window:



The characteristic set drop down list at the top centre of the window is the key to this dialog. It allows you to pick the selected characteristic set. Once selected, the lower part of the window will update to show that characteristic set's trackable characteristics.

Individual trackable characteristics can be selected by clicking any part of them that isn't a field. When selected, their backgrounds turn dark blue/grey:



Click them once more to deselect them. The selection mechanic is used in conjunction with the **Remove** button lower right. Clicking the bottom right **Remove** button will remove all selected trackable characteristics from the current set.

Each trackable characteristic has a number of settings:

The **Name** field describes what the characteristic is tracking. It will be seen in the character, npc, token or campaign asset's details window.

The **Abbrev** field is for the abbreviated name. This will appear in all tooltips.

Colour is used to define the colour of this trackable characteristic's health-bar. The colour can be changed by double clicking the coloured box or by clicking the **...** button.

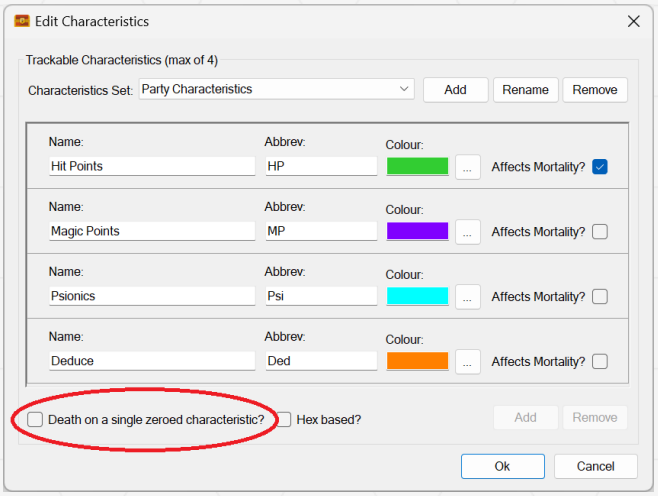
Affects Mortality? Determines whether this field is used for mortality calculations. See below.

Mortality Calculations

Sojour calculates whether an entity is dead by analysing all of its trackable characteristics that are marked as **Affects Mortality**.

By default, all of the characteristics that affect mortality, have to be zero or less for Sojour to determine that the entity is dead.

However, the characteristic set has a tick box at the bottom called **Death on a single zeroed characteristic?** :



When this setting is ticked, any *one* of the characteristics that affect mortality need be zero or less for the entity to be counted as dead.

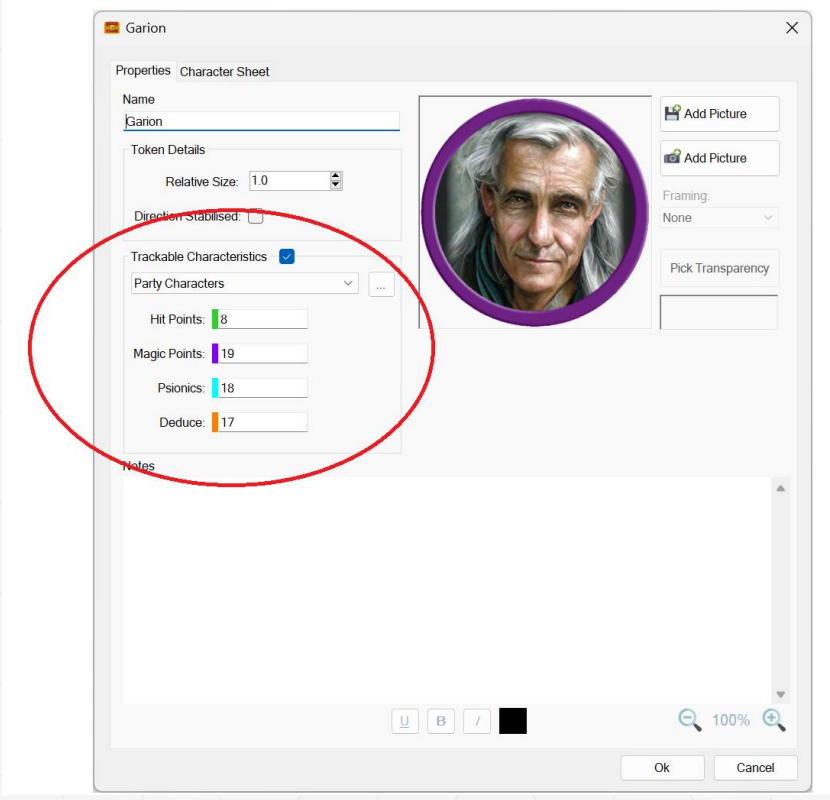
By default Sojour will hide dead entities from the maps and initiative tracker, though this can be turned off in Sojour's settings.

Hex based?

A characteristic set can be made to use the hexadecimal numeric system rather than the decimal system. This has been added as some table top role playing games use hexadecimal.

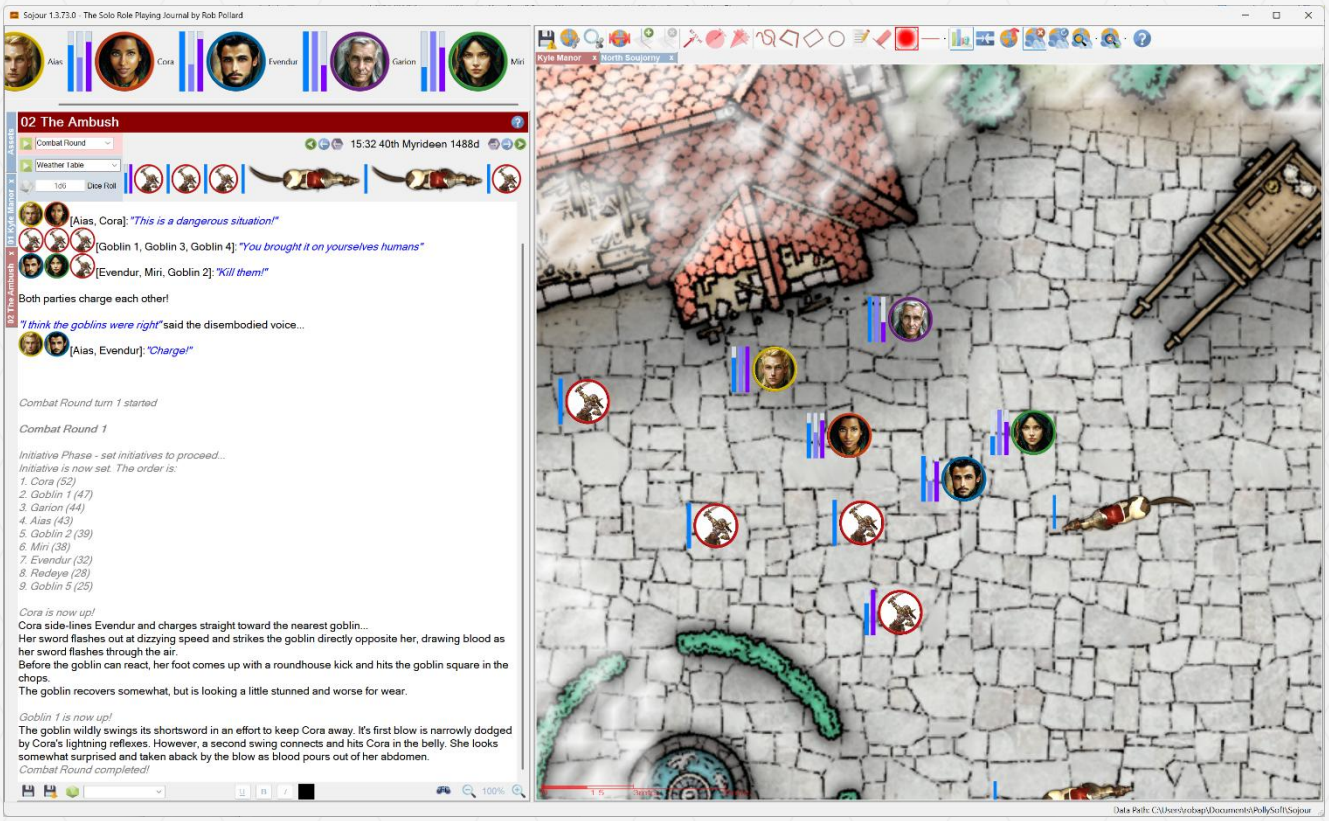
Characteristic sets in the rest of Sojour

Characteristic sets will appear in an entity's details window with matching colour codes:



From here individual sets can be picked or edited. Characteristics can also be enabled or disabled for the entity and individual characteristics can be disabled by setting them to zero. The above screen will be covered in more detail within the Token details section.

Once an entity's characteristics have been setup, health-bars for those characteristics will be visible on the portraits and on the map:



The health bar values can be altered by hovering the mouse over them and using the mouse wheel. The characteristic that is altered is determined by any modifier keys pressed whilst using the mouse wheel:

Modifier key pressed when using mouse wheel	Result
No key pressed	The first characteristic is altered
<Ctrl> key pressed	The second characteristic is altered
<Shift> key pressed	The third characteristic is altered
<Ctrl><Shift> keys pressed	The forth characteristic is altered

Characteristics are numbered from left to right, so the first one is the left most and the forth one is the right most.

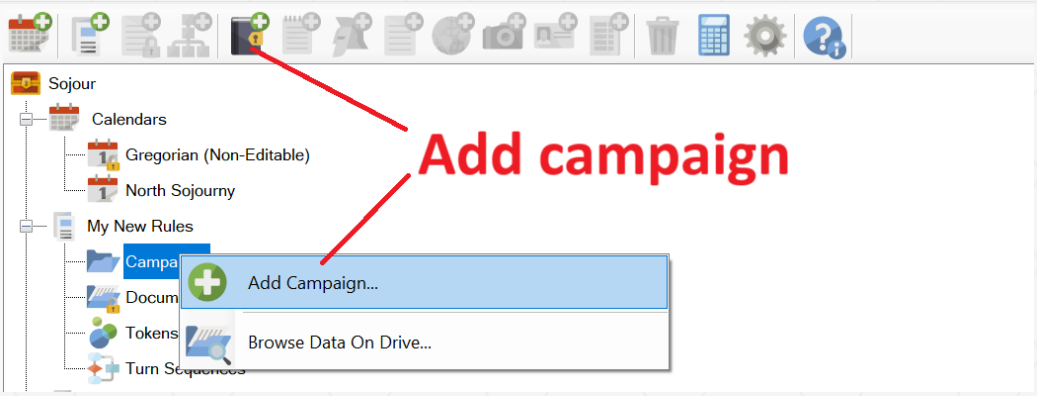
Health bars are percentage based rather than numeric based. For example a character with 12 hit points and 5 magic points, both of which are at that character's maximum values, will have those characteristics represented as two equal length health bars – both rendered as full.

If the hit points in the above example were to drop to 6, the hit point health bar will be 50% filled in, whilst the magic point health bar will still be at 100%.

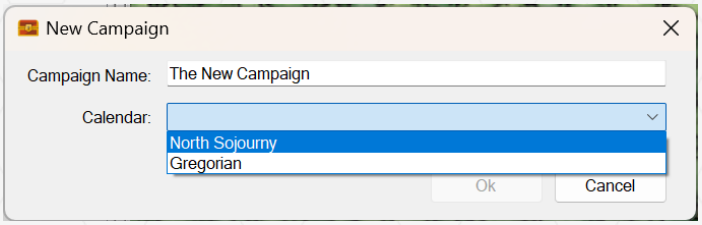
Campaigns

A campaign represents an ongoing story in a coherent time and place. Every campaign is linked to a calendar which is used to keep track of time within that campaign.

A new campaign can be added in one of two ways. Either by right clicking a campaign folder to bring up the context menu, or by clicking a campaign folder and clicking the **Add Campaign** button in the main toolbar:

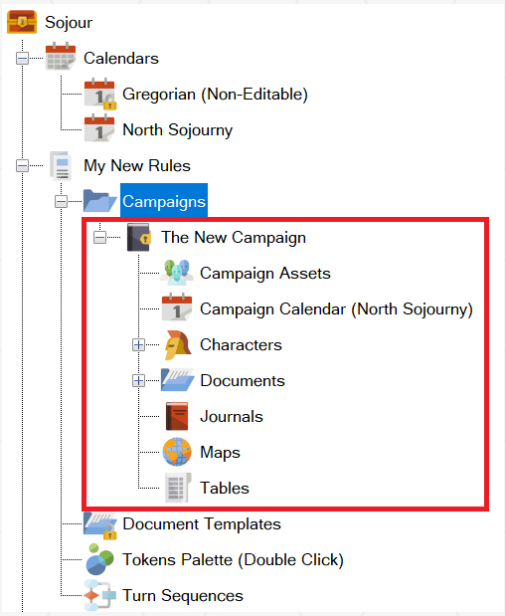


Once you choose to add a new campaign, the **New Campaign** window will appear:



In it you specify the campaign's name and the calendar it is going to use to keep track of time. The calendars shown in the drop down are the ones you have defined under the **Calendars** folder of the **Assets Browser**.

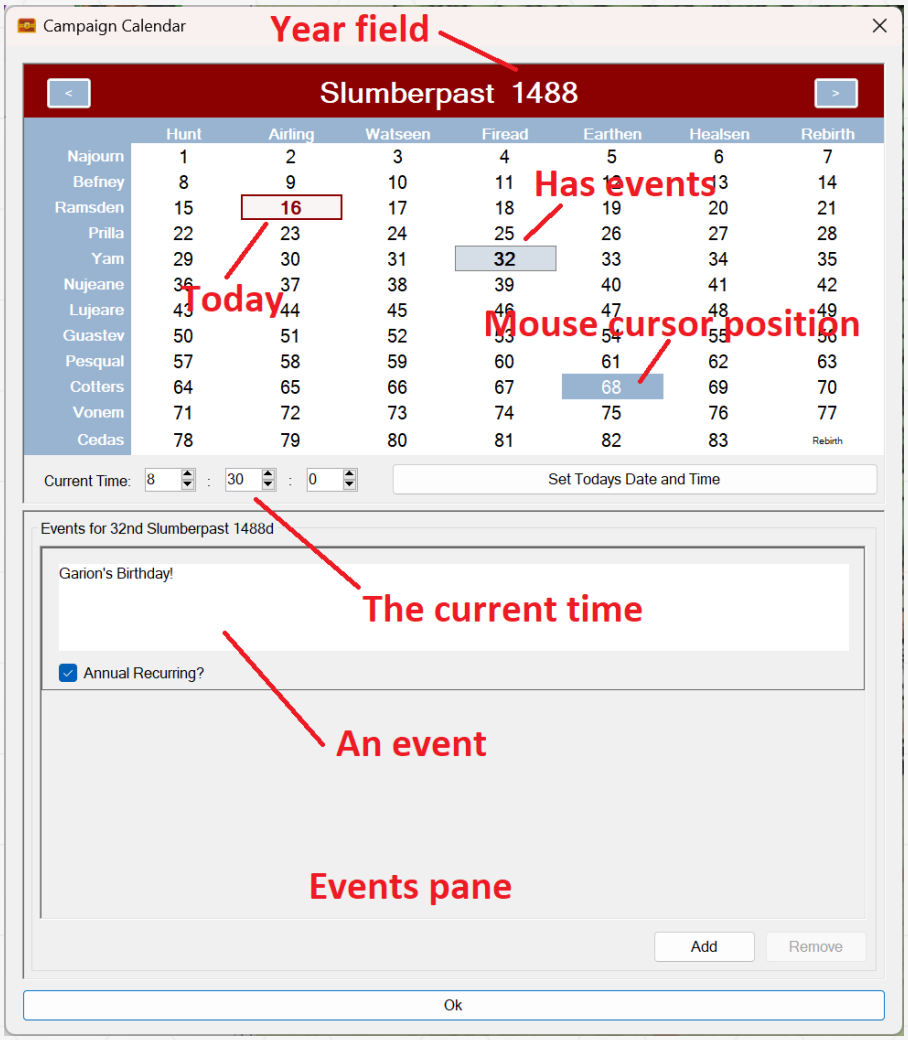
Adding a new campaign results in a number of new folders being created under it:



These folders are defined as follows:

1. **Campaign Assets**: Used to store tokens and their optional single related document that represent assets for the campaign, but not a character or a monster. A good example of a campaign asset would be a star ship in a Sci-Fi RPG. More information can be found in the Tokens, characters and campaign assets section.
2. **Campaign Calendar**: This is an instance of your chosen calendar and is unique to this campaign. It keeps track of time and it allows you to book events, that when triggered, appear in the journal. More details can be found both in the Calendars & Campaign calendar and events sections.
3. **Characters**: These are the main characters in your campaign. Note, that unlike other VTTs, the intent is that you store both party characters and other major campaign characters here too. Active characters appear in the main toolbar. More information can be found in the Tokens, characters and campaign assets section.
4. **Documents**: This folder is used for storing documents related to your campaign. These will be in a PDF or RTF format. They can be imported from the filing system, created locally, or created from a document template on the ruleset. Quick access character sheets can also found under this folder. More information is available in the Documents section.
5. **Journals**: Are the beating heart of Sojour. You will spend most of your time in the journals writing out the narrative of your campaign as it unfolds. More information can be found in the Journals section.
6. **Maps**: Whilst journals deal with the narrative, the maps deal with the place. Maps can be imported from existing images, or can be taken from a screenshot using Sojour's built in functionality or they can be created as blank maps. More information can be found in the Maps section.
7. **Tables**: These are used as static tables to store data pertinent to your campaign or they can be used as lookup tables to trigger events in the journal. Lookup tables can even refer to other lookup tables! More information can be found in the Tables section.

Campaign calendar and events



The campaign calendar is one of the most important aspects of your campaign. It keeps track of the campaign's time and can trigger events in the journal when the correct day arrives.

One can view different days by using the light blue month increment and decrement buttons and by clicking the year field to type in a new year. Note the year field auto-closes after around 1 second of inactivity, so keep typing if you want to keep changing it!

Selecting a particular day will show you the events for that day in the lower events pane – assuming that day has any events.

To set the campaign's date and time, simply pick the year, month and day, select the correct time, and then click the **Set Today's Date and Time** button.

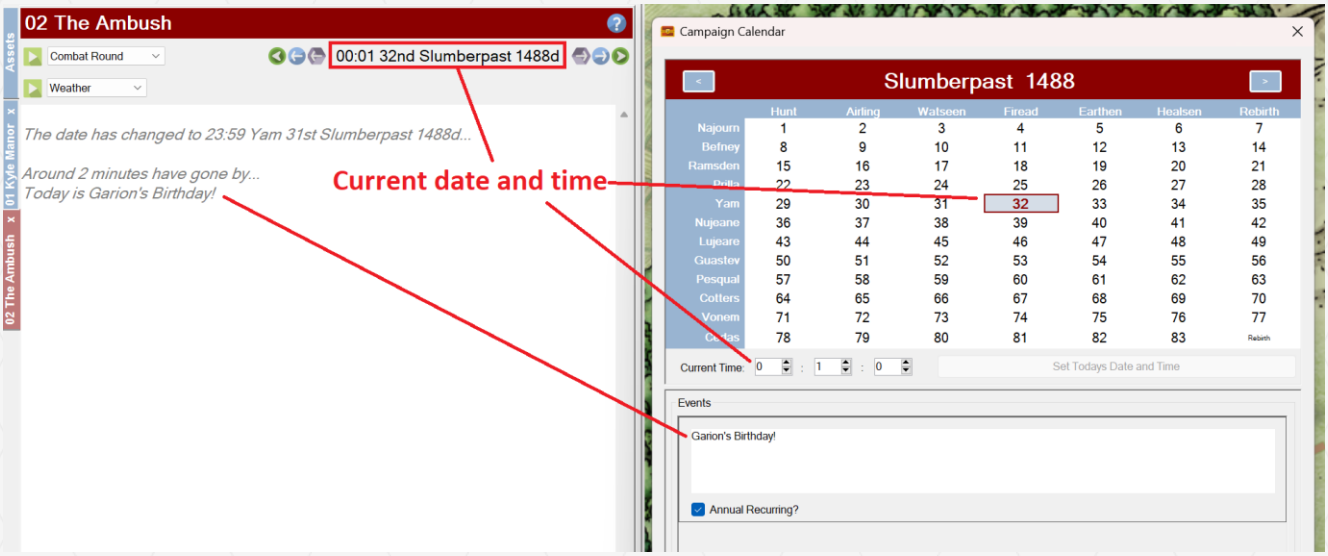
The current day is always highlighted with a red background, red numbering and a red square.

The currently selected day – the one clicked on - has a dark grey border drawn around the day number, which you can see as the 32nd above. (It's partially hidden due to that day also having events.) The currently selected day is also visible on the title for the events pane.

Days that have events have their day number bolded and have a light blue-grey background.

Events in the calendar are trigged at 00:00 of the day they are assigned to and will show up in the journal.

In the screenshot below, time got incremented by 2 minutes which took us onto the 32nd day of Slumberpast which triggered **Garion's Birthday!** event to appear in the journal:



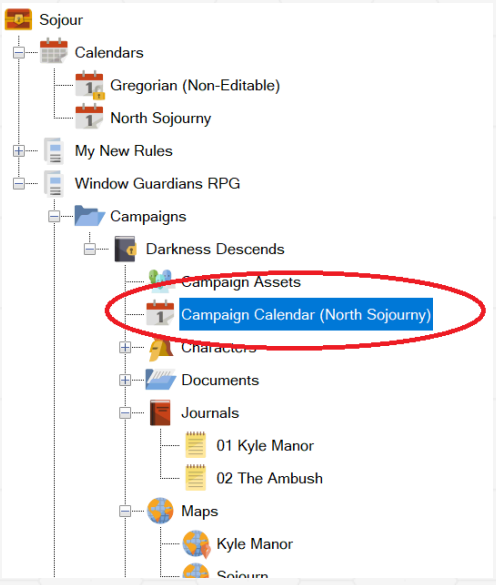
All of the text within the event field will appear directly in the journal.

Triggered events only appear in the active journal. That is, the journal that is open, and currently in view. Sojour used to update all open journals in a campaign, but I found that this approach ended up vandalising already written up journals.

Although the active journal is the only one that gets events updates, all of them, within the same campaign, get the current date and time.

Campaign calendars can be opened in one of two ways:

1. Double clicking the Current Date & Time in the journal as shown in the screenshot above.
2. Or by double clicking the **Campaign Calendar** tree node in the assets view as shown below:



Journals

Journals are the beating heart of your adventures in Sojour. It is within these journals that you will create the narrative that will bring your stories to life.

Warning! A note about keyboard focus. Sojour uses the mouse to determine what has focus. If the mouse is directly over the map, the map will receive all keyboard inputs. If the mouse is anywhere else, the other items that have windows focus will receive keyboard input.

Sojour journals provide a rich set of functionality to assist you with this:

Each of the journal parts are described in more detail below.

Title

The title of the journal. Journals can be renamed in the **Assets Browser** pane by right clicking them and picking **Rename <<Journal Name>>** from the context menu.

Turn Sequences

These are game turns that you can activate. They keep track of what should be happening in a turn, who has initiative and they keep the time updated. You can read more about Turn sequences in their section within this manual.

Npcs (Non player characters)

Npcs are tokens that are not considered main characters. They tend to be your monsters and mobs. These represent the npc's on the currently displayed map. Main characters do not show in the npc bar – they have their own bigger bar top left of Sojour's main window.

Npc characteristics are automatically rolled for when they are dragged onto the map. Npcs are tokens that have been dragged on to the map from the token palette. Their base stats are determined there. More information can be found in the Tokens, characters and campaign assets section.

Hovering the mouse over an npc in the journal's npc bar will place a red ring around that npc on the map. It will also centre the map on that npc if the appropriate options are set.

This works in reverse too. If you hover over an npc on the map, their token will be scrolled into view and highlighted on the npc bar.

The mouse wheel can be used to alter an npc's characteristics when it is hovering over a particular npc in the journal's npc bar. The specifics on altering a characteristic bar can be viewed here.

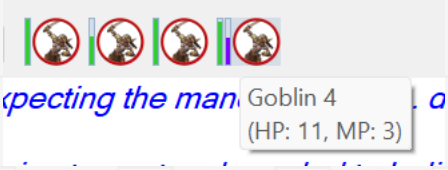
When an npc's characteristics meet the mortality criteria for their characteristics set, they will be hidden from the map and initiative tracker. This behaviour can be disabled in Sojour's settings.

Hidden npc's still appear on the journal npc bar to allow the player to add more damage, or perhaps resurrect them.

Alterations to an npc's health will be seen on the map as floating numbers in the colour of the characteristic that was altered:

Characteristics can be altered by using the mouse wheel over the npc portraits or by placing the mouse cursor over their characteristic bars on the map. You know when you are in the right place on the map as the cursor will change to an up and down arrow as shown above.

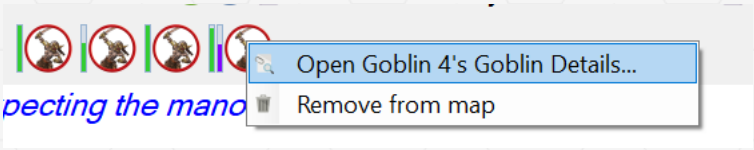
Hovering the mouse over an npc's token on the journal toolbar will provide you with additional details:



These same details are also available by hovering the mouse cursor over the token on the map.

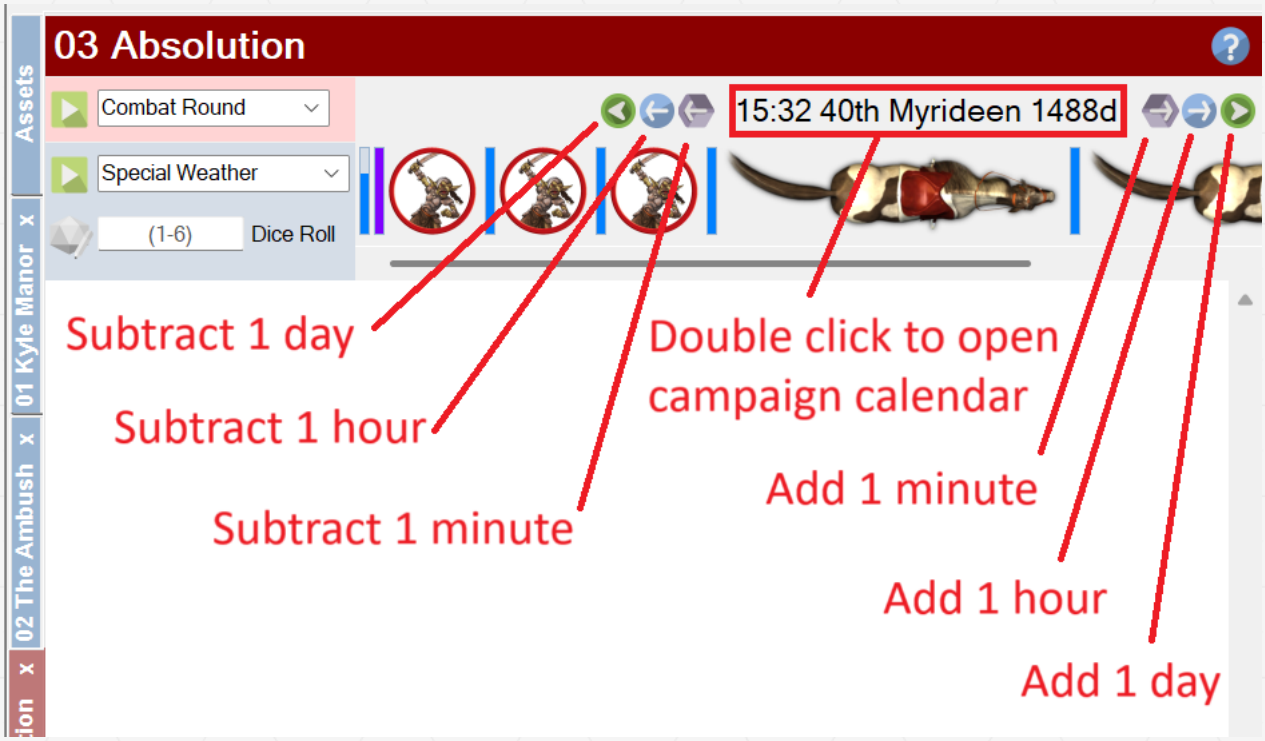
Health bars are percentage bars rather than absolute bars. Varying valued characteristics can all appear the same length provided that their percentages are the same.

Right clicking an npc in the journal will bring up a context menu that will allow you to remove the npc from the map or view its associated document – these actions can also be performed from the map too.



The npc bar will auto show a horizontal scroll bar to allow more npcs to fit than would be allowed by window space.

Calendar and time buttons



Every journal shows the campaign's current date and time in the upper right.

Flanking the date and time are a number of buttons to allow you to easily add or subtract days, hours or minute's from your campaign's calendar.

The user can also double click the date and time to access the campaign calendar. (A hint is provided when the mouse is hovered over the date and time).

There are finer time controls built right into the journal itself by typing various keyboard commands.

All time commands are case sensitive and must be typed in upper case:

Command	Result
+*n*D	Add *n* days to the calendar
-*n*D	Subtract *n* days from the calendar
+*n*H	Add *n* hours to the calendar
-*n*H	Subtract *n* hours from the calendar
+*n*M	Add *n* minutes to the calendar
-*n*M	Subtract *n* minutes from the calendar
+*n*S	Add *n* seconds to the calendar
-*n*S	Subtract *n* seconds from the calendar

For example typing -3D will subtract 3 days from the current campaign calendar.

All journals within a campaign will have their calendars updated. However, only the visible journal will have its journal text updated with the time change. As an aside, the original version of Sojour updated all journals related to a campaign, but this resulted in older journals being vandalised! So we no longer do this!

This is the text that would appear in the open journal after typing -3D:

Around -3 days have gone by...

This text replaces the written command.

The available commands for the journal pane can always be viewed by clicking the **Show journal keyboard shortcuts** button:

Shortcuts help

Clicking the shortcuts help button will result in this window being opened:

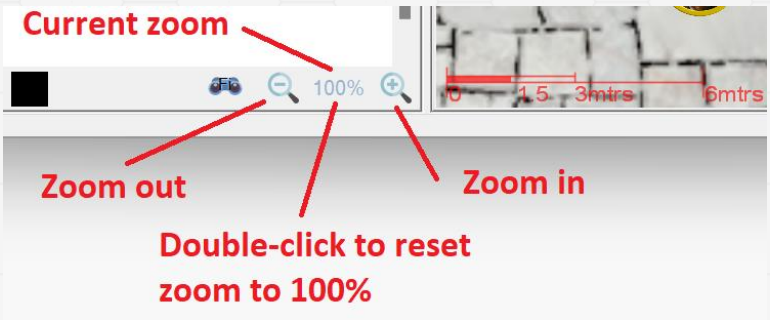
Key	Action
<Ctrl> 1..n	Starts or stops the nth character conversing
<Shift Ctrl> 1..n	Starts or stops the nth npc conversing
<Ctrl> t	Starts anonymous conversation
"	Terminates a conversation
<Ctrl> d	Reset font to regular black
<Ctrl> i	Toggle italics
<Ctrl> u	Toggle underlining
<Ctrl> b	Toggle bolding
[<Dice Expression>]	Rolls the dice expression and replaces the expression with the result
[[<Dice Expression]	Rolls the dice expression and replaces with dice roller text
{ <Dice Expression> }	Rolls the dice expression and replaces the expression with a detailed result
{{ <Dice Expression> }	Rolls the dice expression and replaces with detailed dice roller text
+SD	Insert the short version of the current date
+LD	Insert the long version of the current date
+T	Insert the current time
+DD	Insert current date with no time
+nnS	Add nn seconds to the current time
-nnS	Subtract nn seconds from the current time
+nnM	Add nn minutes to the current time
-nnM	Subtract nn minutes from the current time
+nnH	Add nn hours to the current time
-nnH	Subtract nn hours from the current time
+nnD	Add nn days to the current time
-nnD	Subtract nn days from the current time
<Ctrl> s	Save the journal. Note: The journal does autosave on shutdown anyway
Right click word	Show spelling. Green: Good. Red: Incorrect with suggestions
<Ctrl> Mouse Wheel	Zoom in or out. Zoom settings are saved
Mouse wheel healthbar	Raise or lower the npc's first characteristic
<Ctrl> Mouse wheel healthbar	Raise or lower the npc's second characteristic
<Shift> Mouse wheel healthbar	Raise or lower the npc's third characteristic
<Shift> <Ctrl> Mouse wheel healthbar	Raise or lower the npc's forth characteristic
<Ctrl> z	Undo - Undo change - there are 5 levels of undo
<Ctrl> y	Redo - Redo an undone change

Close

This window will stay on top and can be dragged anywhere on your desktop. The next time you re-open it, it will appear where you last left it. The window is also modeless, so you can continue to operate Sojour whilst it is open.

All the keyboard shortcuts and commands associated with Sojour's journal panes are shown here. This window makes a great aide memoire when learning to use the system.

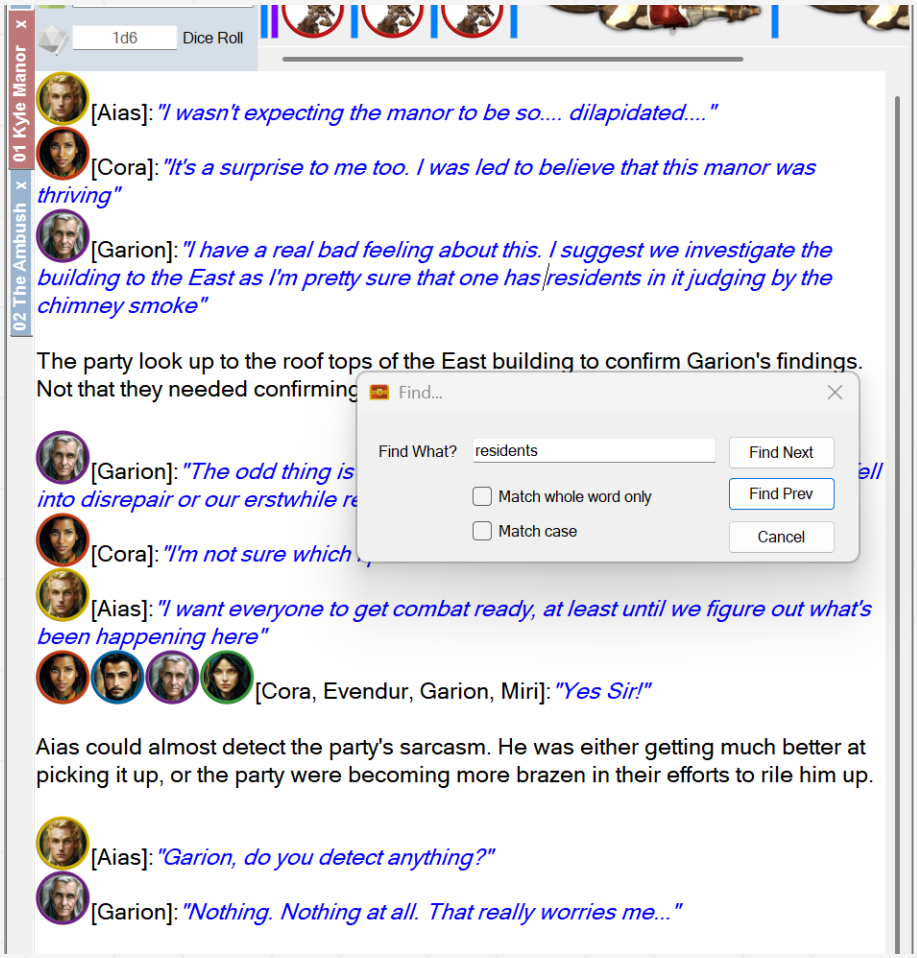
Zoom



Each journal can be independently zoomed in or out to aid in readability. Sojour remembers each journal's zoom settings, so when it is next opened, it will open with the last used zoom setting.

The zoom setting can be changed by using the buttons to lower right of the journal or by using the <Ctrl> Mouse-wheel combination.

Find



The find function works just like any other windows find. It allows you to find text in the currently open journal.

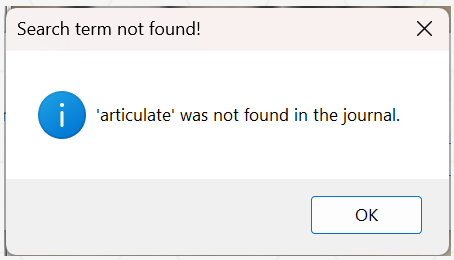
When opened, the find text will default to the word that currently has the caret within it – though you can overwrite this.

The find window is modeless. This means you can drag it anywhere on your desktop and continue working on the journal.

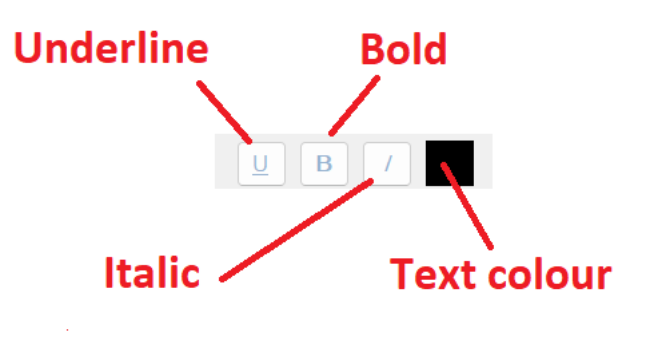
The find function does wrap around, so once it reaches the end of the document it will wrap around to the beginning.

You can opt to match on precise casing or just the word only by ticking the appropriate check boxes.

If the search term is not found the user will be presented with this dialog window:



Font buttons



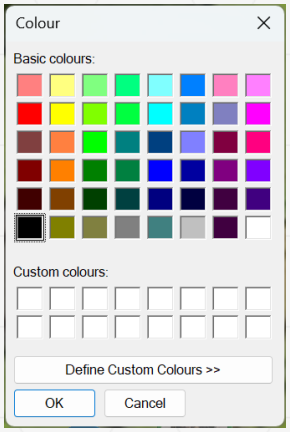
The font buttons serve two purposes:

1. They allow you to change the current font style
2. They show you what the current font style is at the cursor position

A selected option will invert its colour to white on blue:

In the above screenshot the font at the cursor position is blue, underlined and italic.

Clicking the text colour will display a colour picker dialog from which one can choose a standard colour or create a new one with the tools in the dialog:



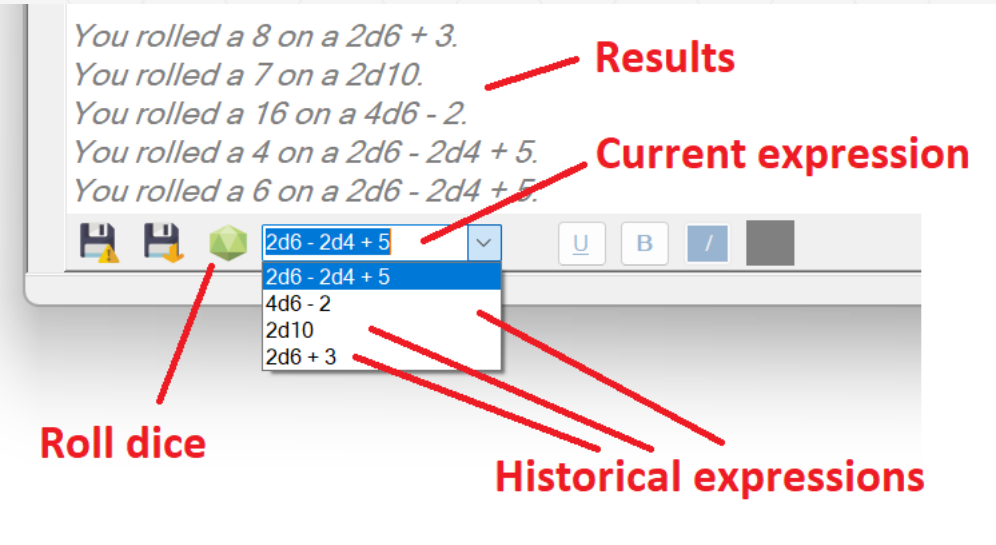
The system always auto-switches to the font where the cursor is located. This allows one to quickly and efficiently edit existing text without having to change fonts to match it.

<Ctrl> D or right clicking any of the font buttons will set the font back to default with a black colour. This is handy when you want to quickly revert back to a standard font without having to open the colour dialog.

Dice Roller

Each journal has two built in dice rollers, one at the bottom of the window and a built in one using typed commands. Both function a little differently and are described below.

Journal dice roller

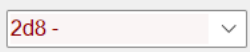


The journal dice roller consists of a button to roll the dice and a field to enter in the dice expression. Rolling the dice will cause the results to appear in the journal as shown above.

The journal dice roller stores a history of the last 10 dice expressions you rolled for that journal. This list is ordered so that the most recently used dice expressions appear toward the top of the list. If an 11th dice expression is created, the oldest historical one is dropped from the list.

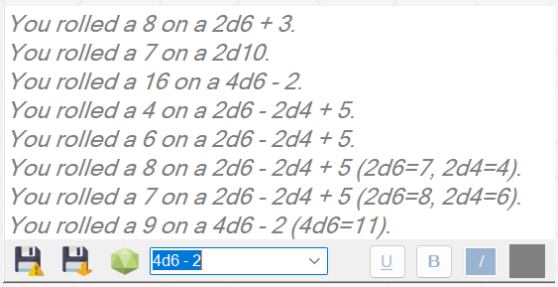
Each journal stores its own independent dice history.

The dice expressions can be as simple or as complex as you like. The expression will change colour to red if it is an invalid expression as shown below:



In this case it is invalid because there is a mathematical symbol but no number or dice expression following it. This feedback is in real time.

Selecting **Show Detailed Rolls** from Sojour's Settings will display a more detailed breakdown of the dice rolls:



The bottom three rolls were rolled with **Show Detailed Rolls** switched on. **Show Detailed Rolls** is off by default, but Sojour will remember whichever option you decide to pick.

Intrinsic journal dice roller

The intrinsic dice roller works a little differently from the one above. It works by typing in a dice expression directly into the journal, once the expression is completed, Sojour will roll the dice and then replace your expression with the result.

The intrinsic dice roller is not affected by the global **Show Detailed Rolls** setting because there are two intrinsic rollers – one for standard results and one for detailed results.

Both operate a little differently.

The journal commands are:

Command	Result
[*Dice Expression* **]**	Roll the typed expression and replace it with the result.
{ *Dice Expression* **}**	Roll the typed expression and replace it with a detailed result.

For example, typing **[2D6 + 2d8]** and **{4d6 – 2d6}** in the journal results in the following output:

17

5 (4d6=14, 2d6=9)

The intrinsic dice roller can be made to display the same output as the journal dice roller at the bottom of the window. Simply double up the first bracket in both modes.

For example, for detailed rolls typing:

{{ 2d6 + 2d4} will result in *You rolled a 14 on a 2d6 + 2d4 (2d6=8, 2d4=6).*

Which is in exactly the same format as the journal dice roller.

It should be noted Sojour was never designed to have dice rollers. The original intent was that users would use their physical rules with physical dice and simply use Sojour as a journal and map. However, user feedback made it clear that a dice roller would be most useful!

Export

Clicking this button allows the user to export the journal as a Rich Text Document that can be opened in Word or WordPad.

Internally, all journals are stored as Rich Text Documents. These can be viewed by right clicking the journal in the **Assets Browser** pane and selecting **Browse Folder…**

Save

The save button was added as a means to manually force Sojour to save the current journal. The same thing can be done with <Ctrl> S.

However, Sojour does automatically save your journal every time you close it or Sojour itself. In addition, Sojour may automatically save your journal periodically in the background, based on your Sojour Settings.

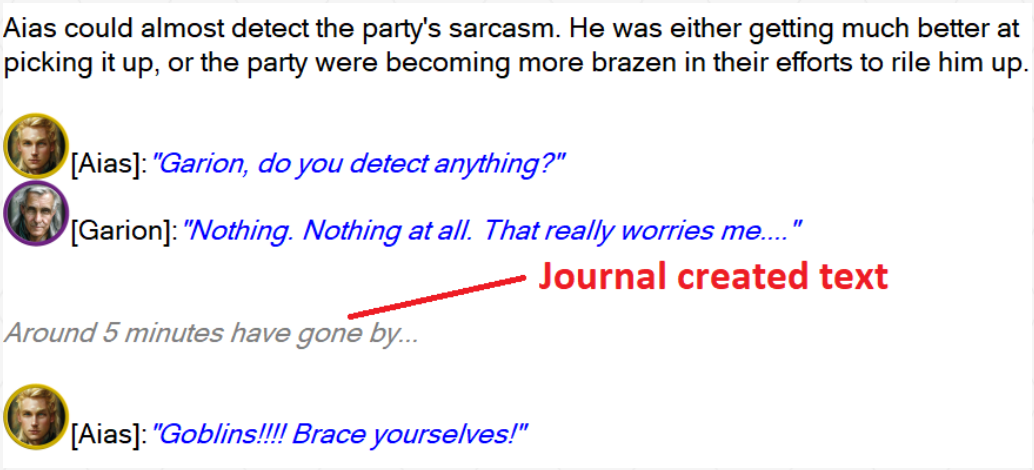
The save button will have a warning icon overlaid on it if there is unsaved work: 

Saving the journal will remove the warning icon.

This feature was added as it is acknowledged that computers do sometimes break or crash, so it is worthwhile being able to periodically perform a manual save to safeguard work.

Journals can also be saved by using the save button on the map tool bar.

Journal created text



The journals display all text created by itself in grey italics, this is to distinguish its text from user typed text.

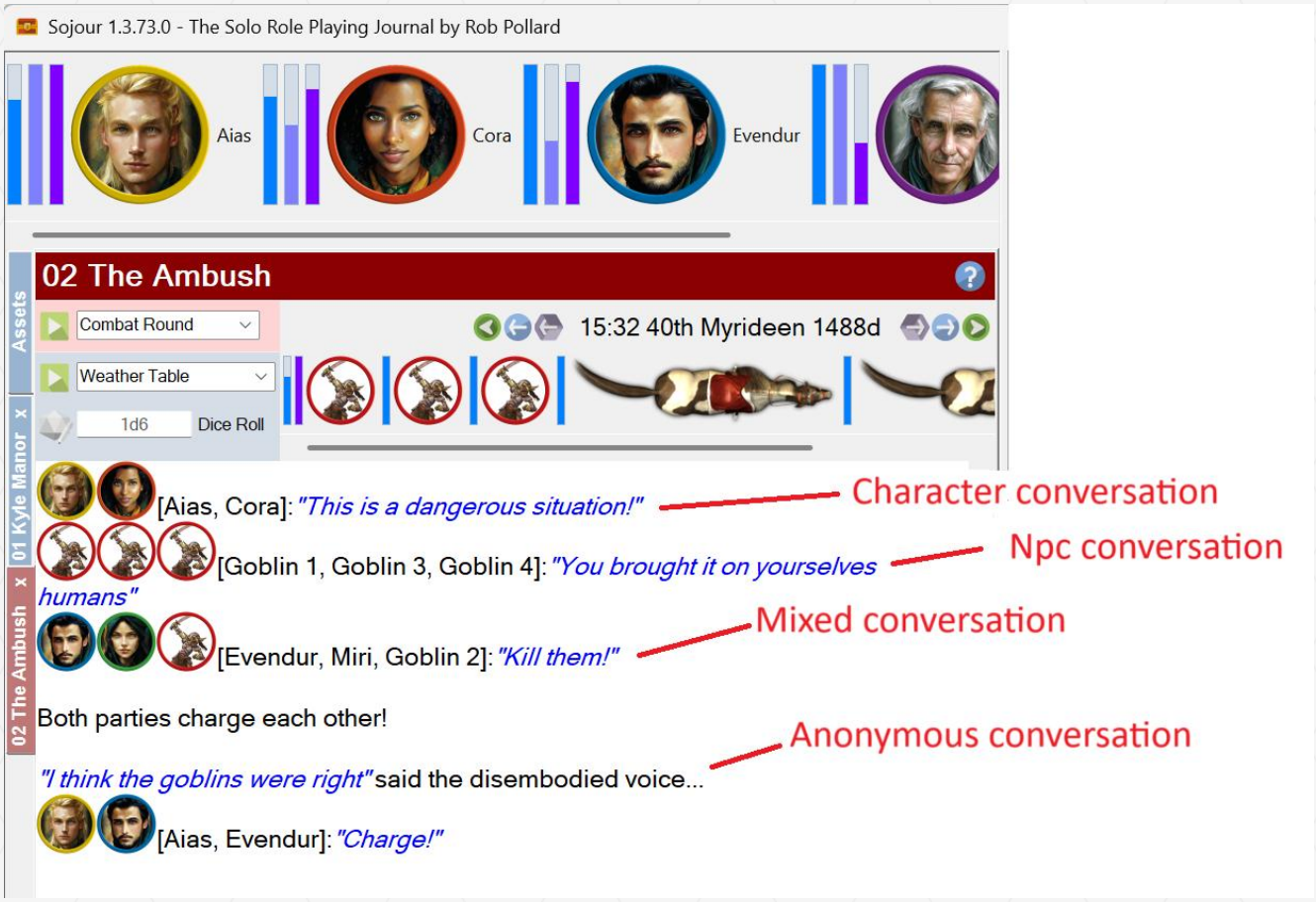
Journal inserted text is editable, just like all other text.

Typical automated entries that you will see Sojour adding are:

1. Dice Rolls
2. Time changes
3. Events from the campaign calendar
4. Lookup table results
5. All output from a running turn sequence including initiative

Sojour inserts text into the currently visible journal only, no matter how many other journals are also open.

Conversing characters



Sojour provides a simple to use mechanism to allow you to easily put conversations into your journal.

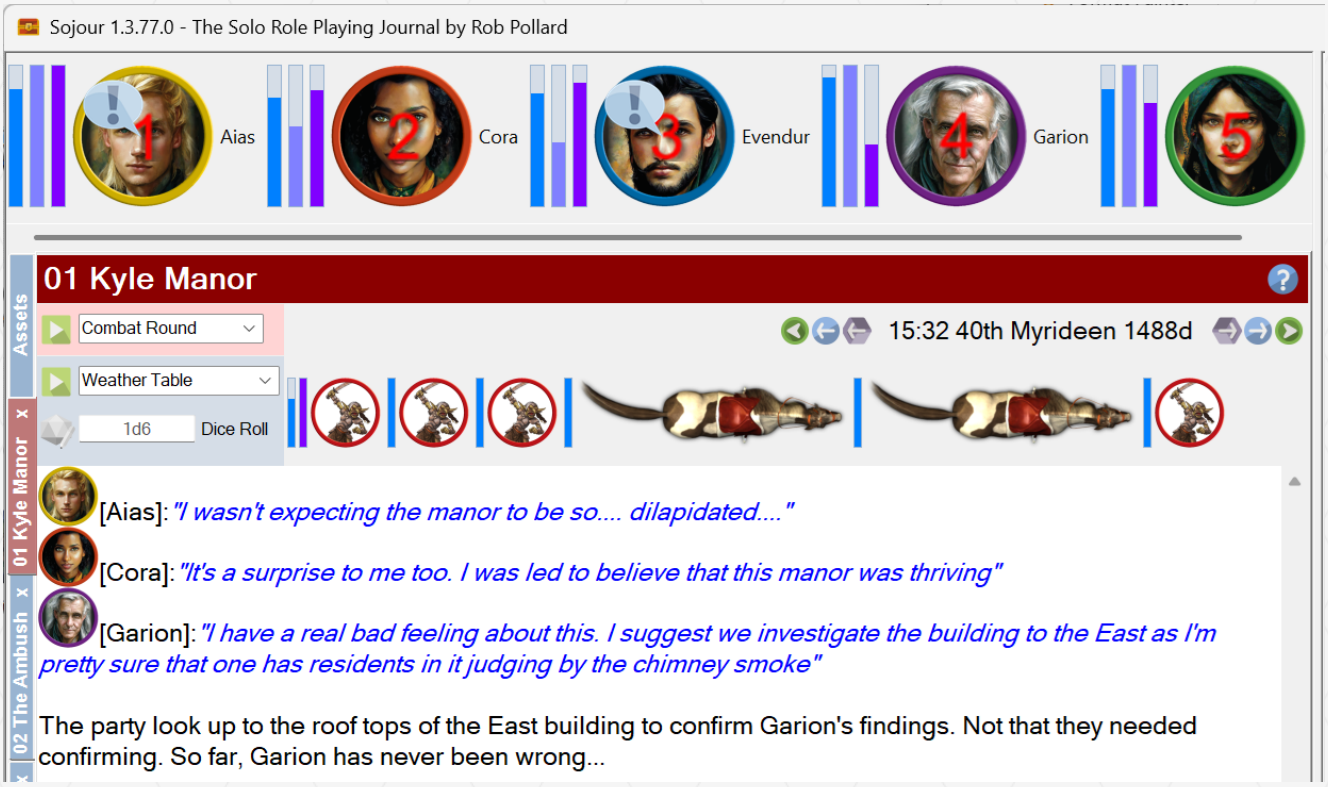
Conversations are initiated / toggled and ended with the following keyboard shortcuts:

Command	Result
<Ctrl> *n*	Toggle character *n* on or off in a conversation.
<Ctrl><Shift> *n*	Toggle NPC *n* on or off in a conversation.
<Ctrl> t (or T)	Start an anonymous conversation
"	End conversation

Conversations are initiated by using the keyboard shortcuts shown above. In addition conversations can be also be initiated/toggled by <Shift><Ctrl> clicking a token directly on the map.

A conversing character or NPC that is selected again, either using the keyboard shortcuts or the map, will be removed from that conversation. In effect character/npc selection acts as a toggle.

The *n* refers to the portrait position from left to right:



Sojour helpfully displays the position numbers on all portraits when the <Shift> or <Ctrl> keys are used and a journal is in focus. Once a conversation has been initiated, the numerics will stay up until the conversation is ended. Otherwise, releasing the <Ctrl> and <Shift> keys will be enough to toggle the numerics off.

Conversing characters and Npcs have a speech bubble overlaid on their portraits and their portrait appears in the journal itself. As noted above, characters and Npcs can be toggled into and out of a conversation simply by selecting the same entity again.

The conversations between npcs and characters can be mixed. Both can be included or removed from the conversation by using the same keyboard shortcuts.

Using Sojour's journal keyboard shortcuts for conversations has a limitation in that only the ten leftmost characters and npcs can converse. However, conversations can also be toggled on and off by left clicking a character or NPC directly on the map whilst holding down the <Ctrl> and <Shift> keys.

Anonymous conversations were added as a shortcut to allow characters that are too minor to have a token to converse in the same way as the other characters and npcs do.

In the example at the beginning of this section, the disembodied voice isn't defined in Sojour as a character or npc; it doesn't even have a token! But we can get the disembodied voice to talk by using the anonymous conversation function with <Ctrl> t.

All conversations are terminated with ".

Once terminated, Sojour will return the font back to the last used font prior to the conversation taking place.

As with all things in the journal, conversations can be edited after the fact.

Deleting any part of an active conversation's prefixed text up to the first speech mark and including any of the portraits will delete all parts of that conversation except player entered text. It will also turn off the conversation.

Anonymous in-flow conversations are likewise cancelled if the first speech mark is deleted.

Associated Tab

This is the tab associated with the journal and is used to bring it into focus. As mentioned earlier, only the in-focus journal gets Sojour's textual updates. However, all journals related to the same campaign keep the same time together.

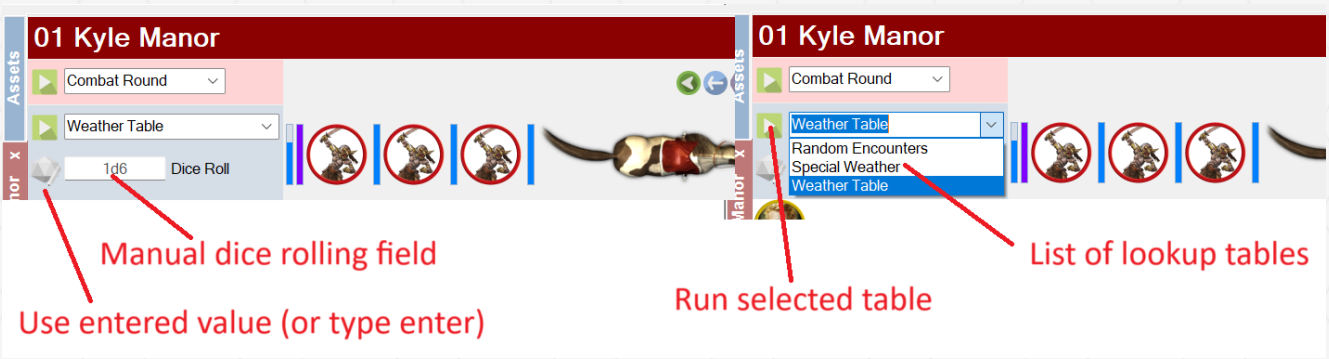
A journal can be closed by clicking the X on the tab. A journal can be re-opened by double clicking it in the assets browser.

Sojour remembers which journals were open between instances. So the next time you start Sojour, it will attempt to automatically open all the journals you left open when you last shut it down.

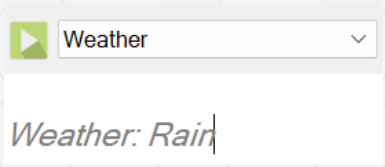
Lookup Tables

Sojour features lookup tables. These are tables that can be rolled on to produce random text in journals.

This is a dropdown list of all the lookup tables for the current ruleset. Only lookup tables are shown in this list. Data tables do not appear.



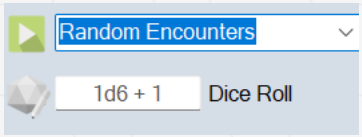
To use a lookup table, simply select the one you want or start typing in the upper field to filter your table results before picking. Once picked, click the green button to run it. You may be prompted to pick some modifiers at this point. But once it is all done you will see the result in the journal. In the example below we trigger the **Weather** lookup table:



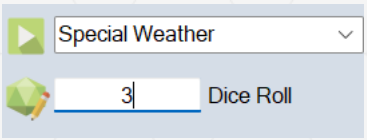
By default, Sojour shows the table name, a colon and then the table results. However, there is a setting in general settings for displaying the rolled result without the table name.

The manual dice rolling field enables you to enter in dice roll results and have the table automatically roll using your results. This is handy for situations where you want to roll your own dice on the tables.

The field tells you which dice to roll and any modifiers that must be applied for the selected table:



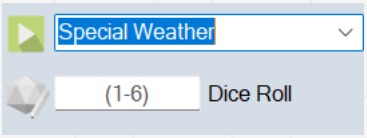
For this table you must roll a 1d6 and add one to the result – don't forget the modifiers – the field will not add them for you. Once you have entered a result the disabled dice button will turn green indicating that you can now use it:



The enter key can be used instead for keyboard users.

Rolls above the maximum range of the table will pick the last entry in the table. Conversely, rolls smaller than the table range will pick the first entry of the table.

If your table is not using dice but a simple range of values, this will be displayed instead:



In this example the range is 1-6 so it is easy to figure out that you can just roll a 1d6. Other tables may need more finessing, or alternately, you can edit the table to support actual dice.

Event tables can be selectively set to be hidden or visible to the journal drop down. If your table does not appear in the drop down list, make sure it is set to be visible and that it is an event table.

Lookup tables are covered in much greater depth within the Lookup Tables section.

Other features

The journal also hosts a number of other features. These are described below.

Spelling

All Sojour journals include spell checking which should pick up your local language from your operating system settings.

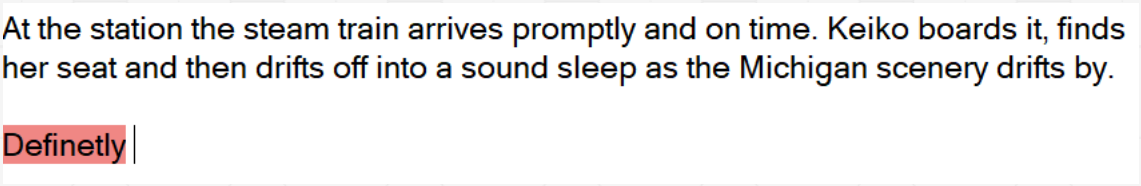
The spelling system has two modes of operation:

1. **Dynamic spell checks:** Checks your spelling as you type. Can be disabled if required.
2. **Static spell checks:** Right clicking a word will show its spelling status.

Dynamic spell checks

This is enabled by default under Sojour's Settings.

Dynamic spell checking briefly highlights incorrectly spelt words in red as shown below:



The highlighting disappears after a few seconds. It's designed to draw the eye to a word you might have misspelt.

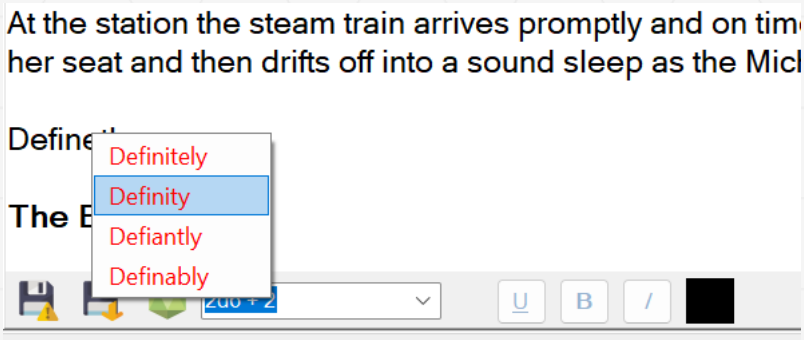
Sojour does not underscore badly spelt words as RPG Journals tend to be filled with many made up words and this can make them look messy. The highlighting system was deemed to be the best compromise.

It's acknowledged that some might find this feature annoying or distracting, so it can be turned off under Sojour's general Settings.

Badly spelt words can then be right clicked on in order to fix them. See the static spelling checks section below.

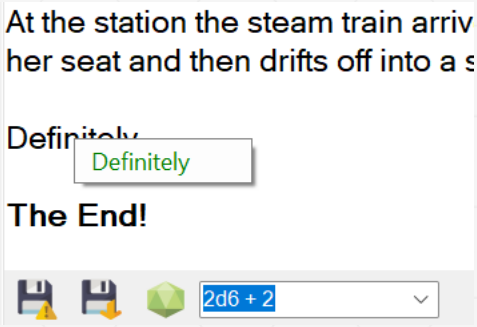
Static spelling checks:

Right clicking on any word will allow you to see that word's spell check status. If the word is incorrectly spelt, the context menu will be displayed in red along with some alternative spelling suggestions as shown below:



To fix the word simply pick one of the options.

In contrast, right clicking a word with good spelling will produce a green context menu with no options:



The static spell checker provides a great way to instantly check on any suspect words.

As noted under dynamic spell checking, Sojour should automatically pick up your local language based on the language that you have set within your operating system. However, I have been unable to test this at present.

Date and time insertion

Sojour allows the current campaign date and time to be easily inserted into the journal using these typed commands:

Command	Result
+SD	Insert the short version of the current campaign date
+LD	Insert the long version of the current campaign date
+T	Insert the current campaign time
+DD	Insert the current campaign date with no time

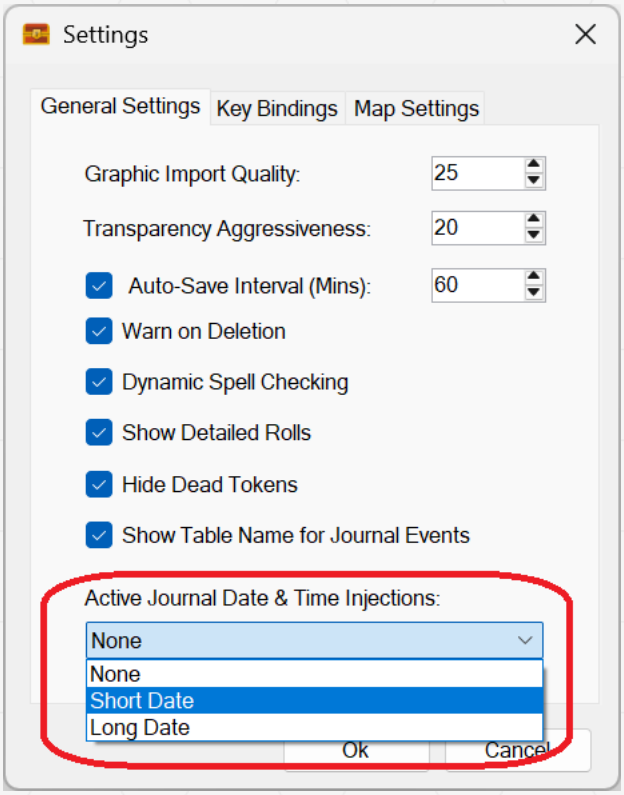
The above commands are case sensitive.

The formatting of the long and short date and time formats are defined in the Calendar that you are using for the campaign.

At the time of writing the Gregorian calendar's date and time formats cannot be altered.

In addition to manual date and time insertion, Sojour will also automatically inject date and time information into the active journal whenever the campaign calendar has had its date or time altered.

What's displayed in the journal is dependent on Sojour's global settings:



The three output types are shown below:

Around 5 minutes have gone by... — None

Around 5 minutes have gone by. It is now 15:42 40th Myrideen 1488d... — Short Date

Around 5 minutes have gone by. It is now 15:47 Nujeane 40th Myrideen 1488d... — Long Date

The Short Date and Long Date refer to the short and long dates defined for your campaign calendar.

Undo-Redo

Sojour journals support 5 levels of undo and redo. Use <Ctrl> Z for undo and <Ctrl> Y to redo.

Tables

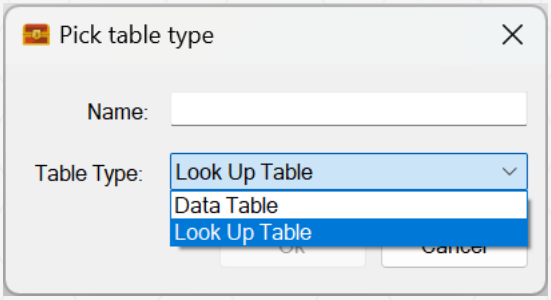
Sojour supports two types of table:

1. **Lookup Tables:** For generating events in the journals
2. **Data Tables:** For storing campaign data

Lookup tables appear in the lookup table's drop down within a journal, data tables do not.

All tables are added to a campaign by right clicking on the **Tables** node of the **Assets Browser** or by using the main toolbar button.

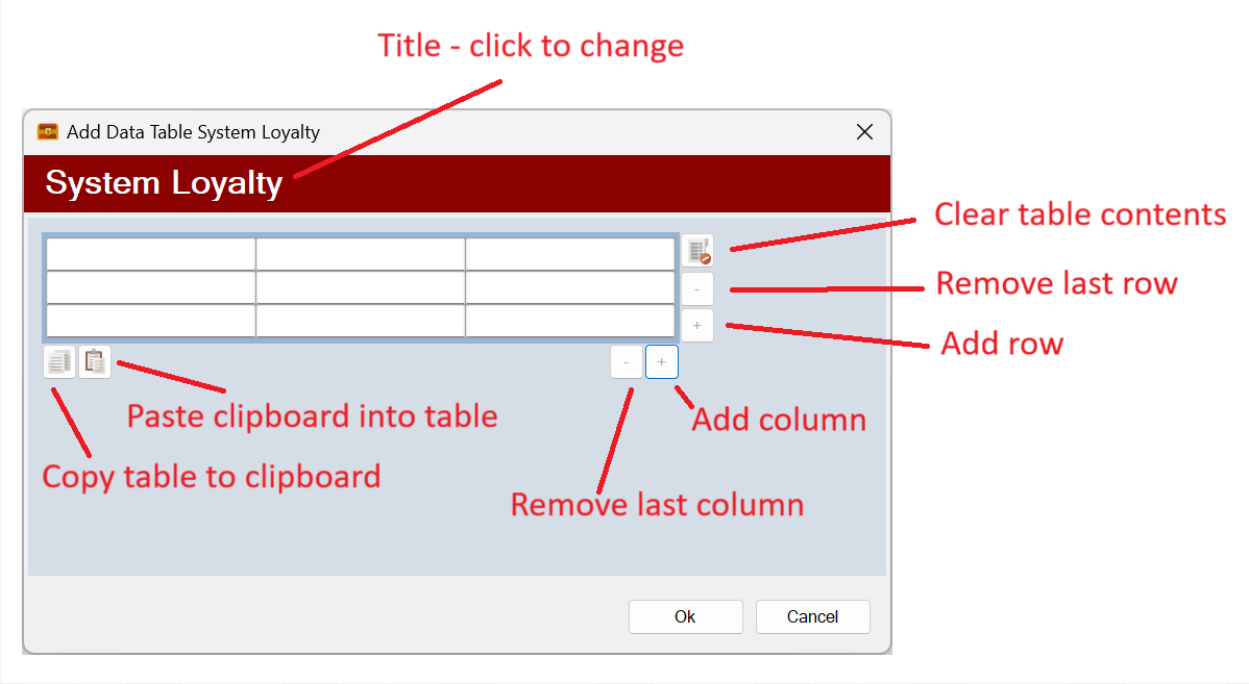
When adding a table you need to name it and then pick the type of table:



We will discuss the two table types below.

Data tables

Data tables are static tables and cannot perform calculations or be used for generating events. Their sole purpose is to store data related to your campaign.



Tables have a number of buttons attached to them that enable you to add or remove columns or rows. Tables must have at least two rows and two columns. If either is at the minimum, the remove button associated with that axis will be hidden.

Table columns can be resized by dragging on the column lines. Row resizing isn't currently supported, but the rows do auto-expand to fit content.

Copy & Paste

Data can be copied and pasted between Sojour tables using the copy and paste buttons.

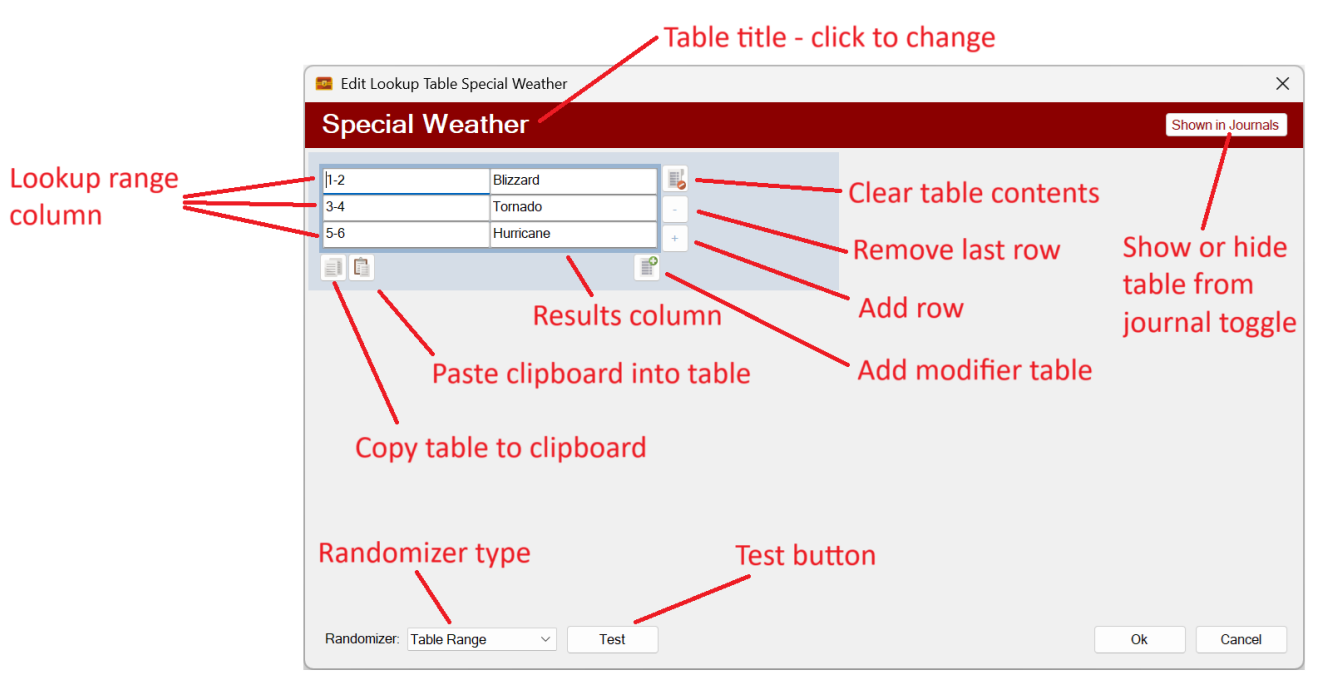
Pasted data is pasted using the currently selected cell as the top left for the pasted data.

If there are not enough rows or columns to accommodate the pasted data, new rows and columns will be automatically added. The exception to this is that lookup tables and modifier tables will only increase the row count, they will never alter the column count which is always fixed at two.

Sojour tables can accept data copied from other sources.

Lookup tables

Lookup tables are used to generate events and text within a journal by rolling on them with virtual dice and generating a result.



Lookup tables only have two columns:

1. **Lookup range:** The leftmost column provides the numeric ranges that must be rolled by the randomizer to activate that row. All ranges take the form of n-n1 or just n for a singular number.
2. **Results column:** The rightmost column provides the text results of the event that will happen should their numeric range be rolled. This column also supports references to other tables, so a roll result could trigger a roll on a different table.

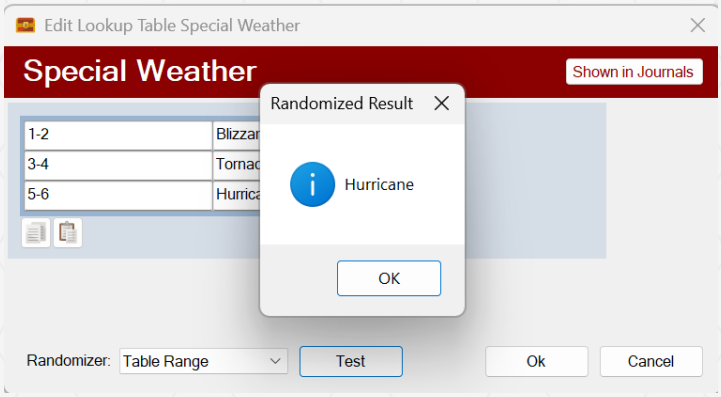
Taking the above table as an example, if the randomizer rolls a 4, the result will be 'Tornado' and this result is displayed in the journal that triggered the roll.

Event tables also have an additional toggle button to toggle their visibility in a journal's table drop down.

Lookup tables support two types of randomizer:

1. **Table range:** This randomizer always rolls for the whole range of the table. It is a single flat roll with an even distribution.
2. **Dice Expression:** This randomizer uses the dice expression provided by the user. Dice expressions can be as simple or complex as needed. Dice expressions do not normally have an even distribution (unless they are composed of a singular die).

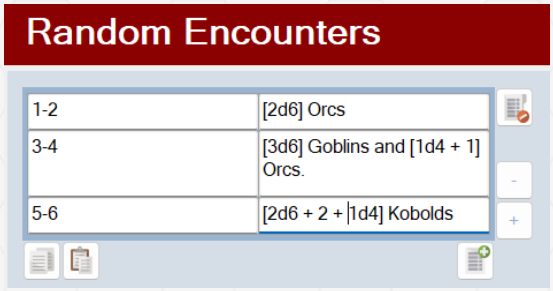
A test button is provided to test your lookup table to ensure that it is working correctly. When pressed, a message window appears showing the result of a roll:



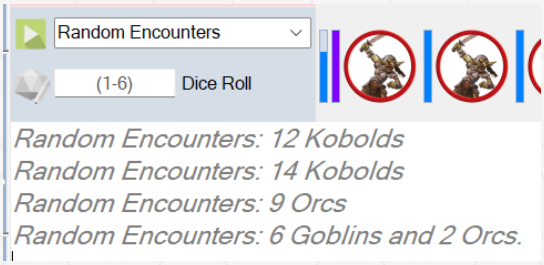
In the above example the test rolled a 5 or 6 and got hurricane as the result.

The table results column can also accept any number of dice expressions. Dice expressions can be entered in one of three formats, the same formats that are supported by the journals. See the journal's intrinsic dice roller for more information.

The screenshot below shows a table with dice expressions:

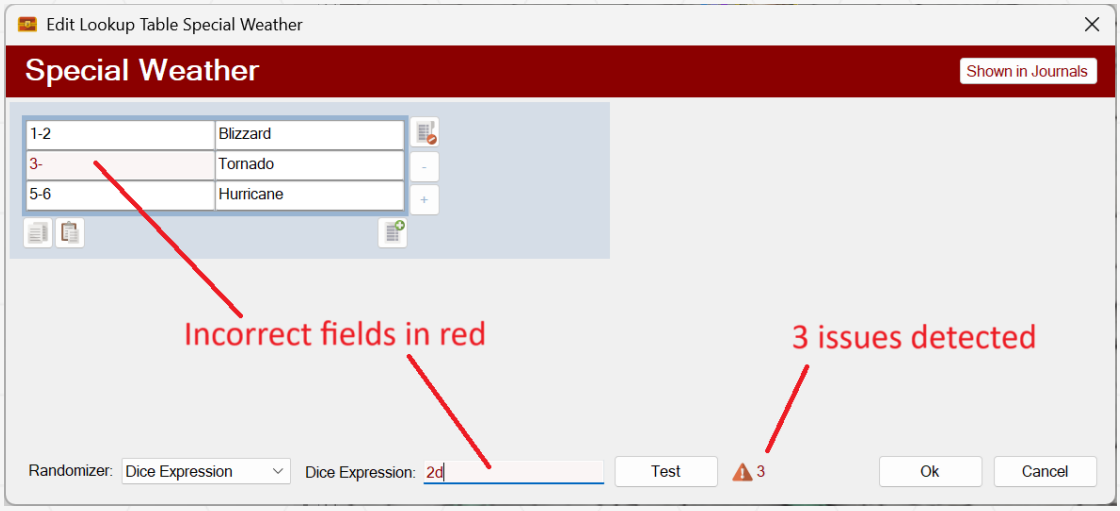


These dice expressions are automatically resolved when the table is rolled for in the journal:



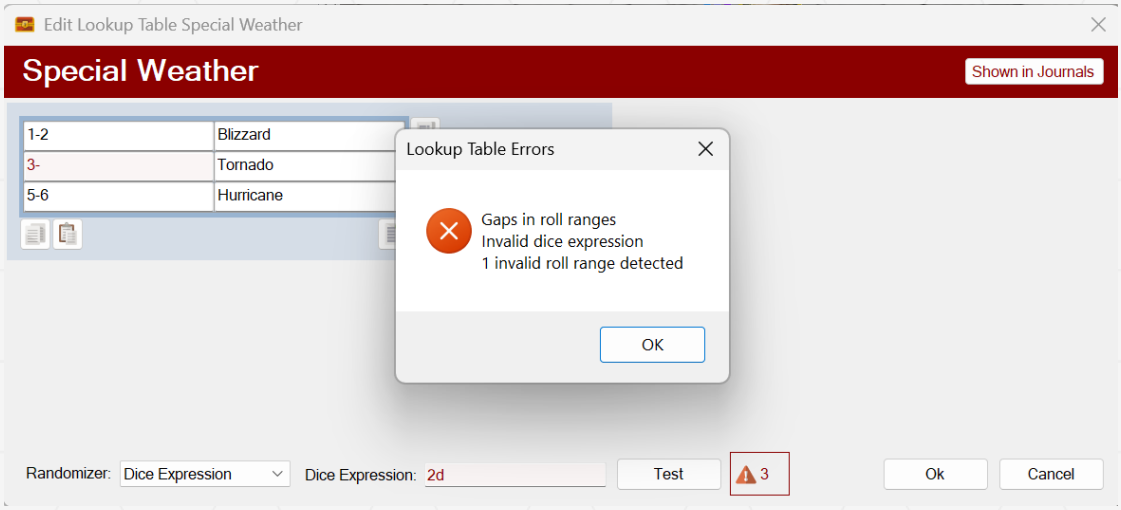
Lookup tables have a real-time validation system that will keep the user informed as to any potential issues. These issues are always treated as warnings, it's the prerogative of the user whether or not to act on them.

In the screenshot below the system has detected 3 issues:



Two of them are immediately obvious as the invalid fields that are highlighted in red.

To get more information simply click the warnings triangle:



Here we have clicked the warnings triangle and it is telling us we have:

1. **Gaps in the roll ranges:** The **3-** row has an invalid roll range, so this table can only parse rolls that are 1-2 or 5-6.
2. **Invalid dice expression:** The dice expression is **2d** but there is no notation to define the actual dice to roll.
3. **1 Invalid roll range detected:** This is referring to the roll range defined as **3-** it is missing the second number required to complete the roll range.

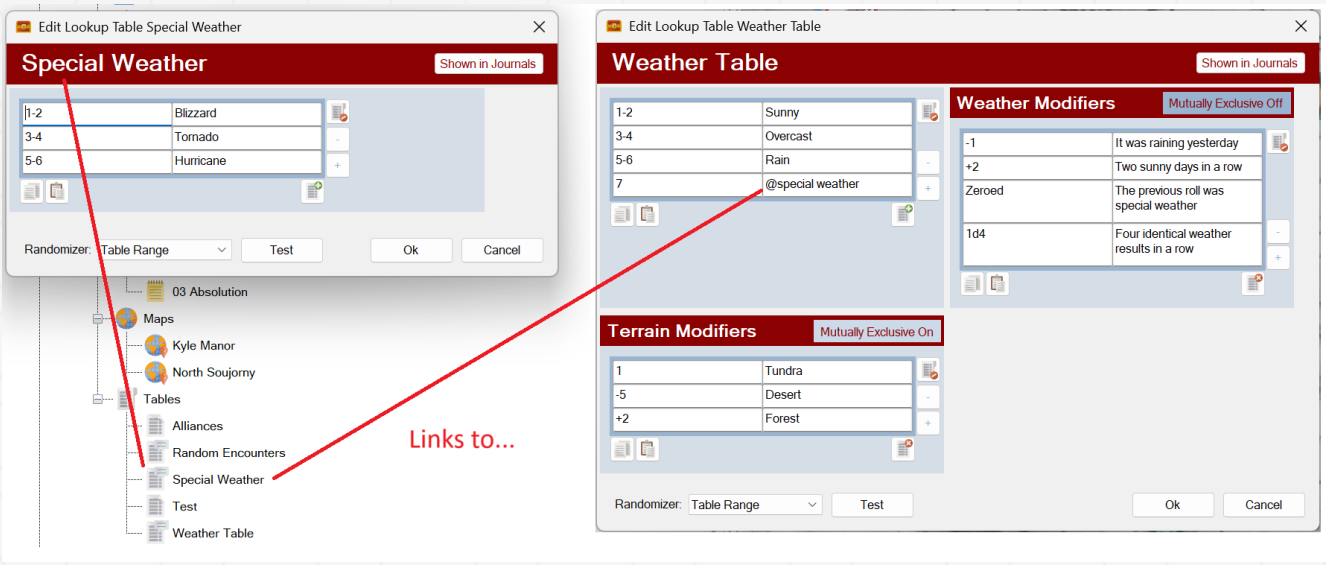
These warnings are never enforced. It's up to the user to address the warnings as they see fit. Only they know which warnings were intentionally created.

Referring to other lookup tables

Lookup tables have a powerful feature whereby a triggered row can trigger a roll on another table. There is no limit to how many tables that can be linked together. You could have a table, linked to a table, which is itself also linked to another table!

If a target table has been deleted or renamed, the activated row will simply display the text in that row's results column rather than jumping to a new table.

Other tables are referred to by using the @ symbol followed by the table name. The table name is case insensitive as shown below:

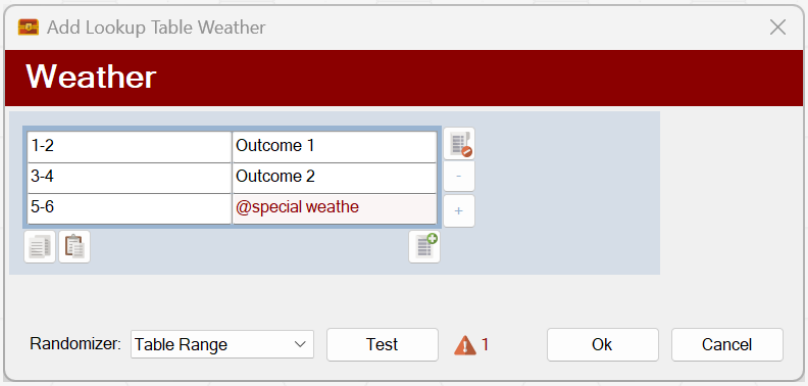


In this example the **7** row of the **Weather** table refers to the **Special Weather** table. So if a **7** is rolled on the weather table, it will trigger a new roll on the **Special Weather** Table.

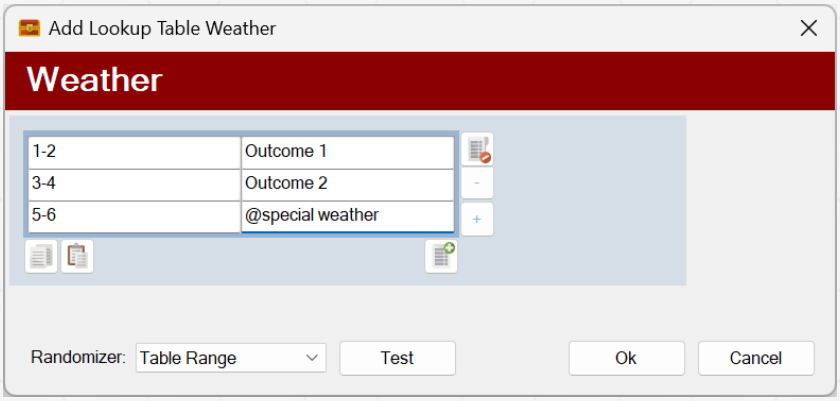
For example, if the system rolled a 7 for **Weather**, this would trigger a second roll on the **Special Weather** table. If that in turn rolled a 5 the result would be 'Hurricane'.

The **Special Weather** table can also be directly rolled on as its own standalone table too.

When a user types a table link, the system checks that link in real-time. The link name must exactly match the table name that you want to link to – though it is case insensitive. If the linked table name doesn't exist, the field will turn red like all other invalid fields:

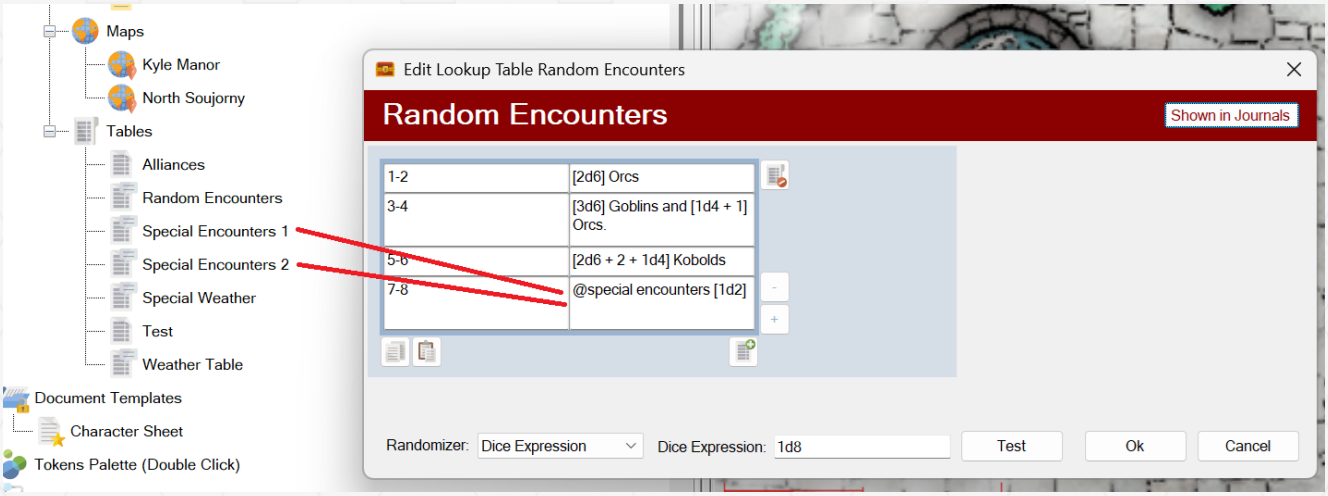


Here, the user hasn't finished typing. 'special weathe' does not exist as a table so it is highlighted in red to let the user know this. In addition, an associated warning is raised in the warning triangle. Once the user completes the sentence by typing the 'r', the field will change to black on white and there will be no warnings displayed:



This mechanism provides instant feedback as to whether a table link is valid or not.

Table links can also include dice expressions that will resolve as part of the table name:



The example above is referring to a table name called:

special encounters [1d2]

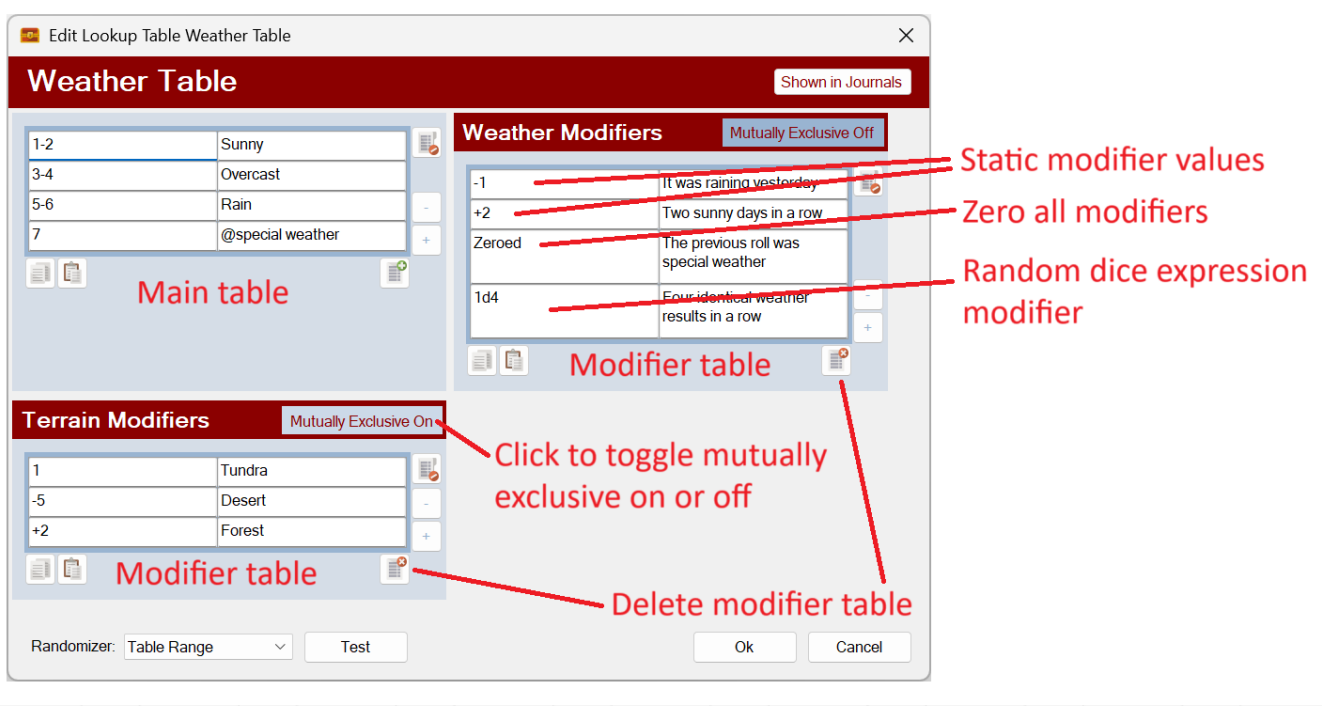
When this entry is run Sojour will roll a 1d2 dice (an impossible dice in real life, but one that Sojour is more than capable of running). The result of the dice roll will be one of two table names:

special encounters 1

special encounters 2

Sojour will then jump to whichever table was rolled!

Modifier tables



Each lookup table can have zero or more modifier tables. These tables provide the user with an opportunity to select modifiers when they roll on that table. These modifiers are applied directly to the roll made on the main table.

Modifier tables appear in the same window as the lookup table that they are associated with. In the above example, the **Weather** table has two modifier tables, one called '**Weather Modifiers**' the other called '**Terrain Modifiers**'.

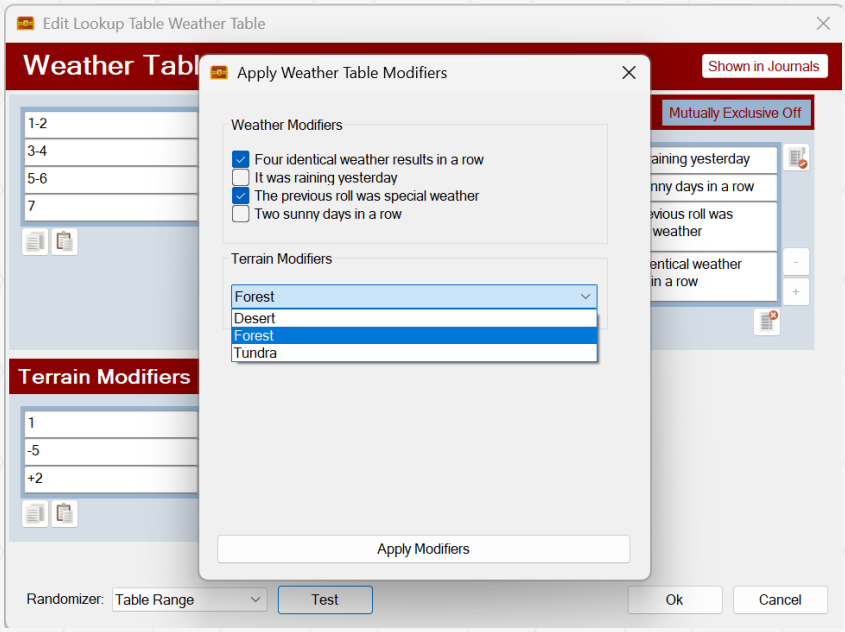
Modifier tables always have two columns.

1. **The modifier**: This is the left most column and is the modifier that will be applied to the roll on the man table.
2. **Modifier description**: This is the right most column and provides descriptive text about the modifier.

There are three types of modifiers:

1. **Static modifiers**: These are simply positive or negative numbers that are directly applied to the roll on the main table.
2. **Zeroed**: This is a special modifier that when selected, zeroes all other modifiers. To use, type in the word **Zeroed** – it is case insensitive.
3. **Dice expression**: These are random modifiers that will roll the dice expression and add or subtract it as a modifier to the main table dice roll.

The mutually exclusive on or off button determines whether the modifiers in that table are mutually exclusive or not. It directly affects how the modifiers are presented to the user when a roll is made on the main table:



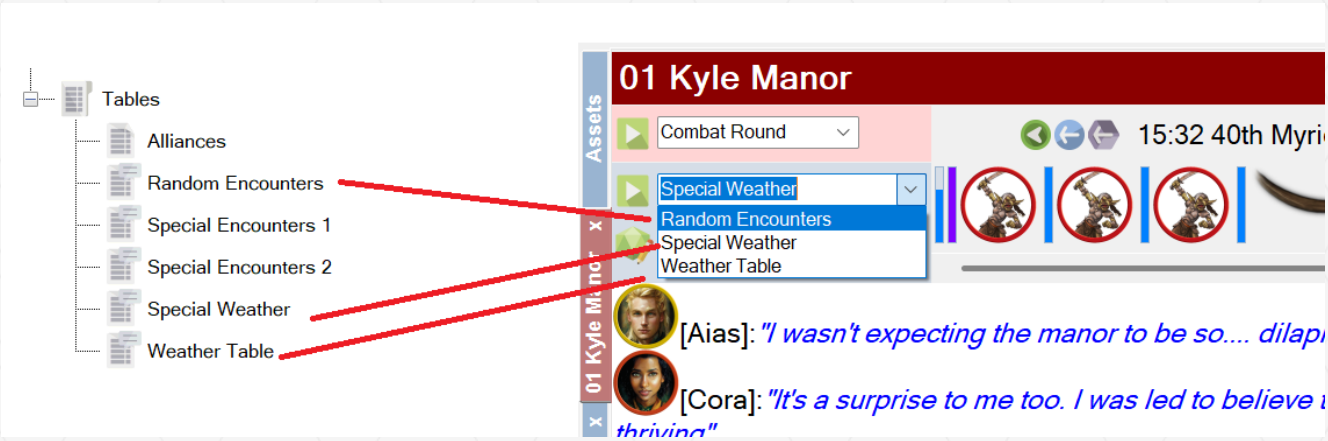
In the example above we have rolled on the weather table. A message window is presented to the user to allow them to pick their modifiers. Lookup tables with no modifiers will not display this window.

The **Weather Modifiers** in this example are not mutually exclusive, so the user can pick any combination of them by ticking or unticking each modifier.

However, the **Terrain Modifiers** are mutually exclusive, so the user is presented with a drop down list from which they can pick only one. Note, that a user *must* pick at least one option on a mutually exclusive modifier table to proceed. If you want a modifier to be optional, it must be set to mutually exclusive off.

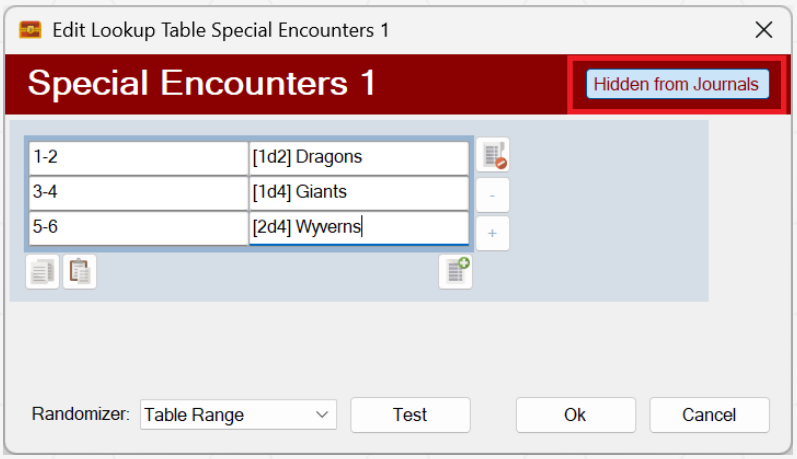
Once all the modifiers are picked, the user simply clicks the **Apply Modifiers** button. This will apply the selected modifiers to the roll made on the main table and then display the result.

All lookup tables are displayed in a drop down within each journal that is part of the same campaign (tables are tied to a specific campaign and are set to be visible):

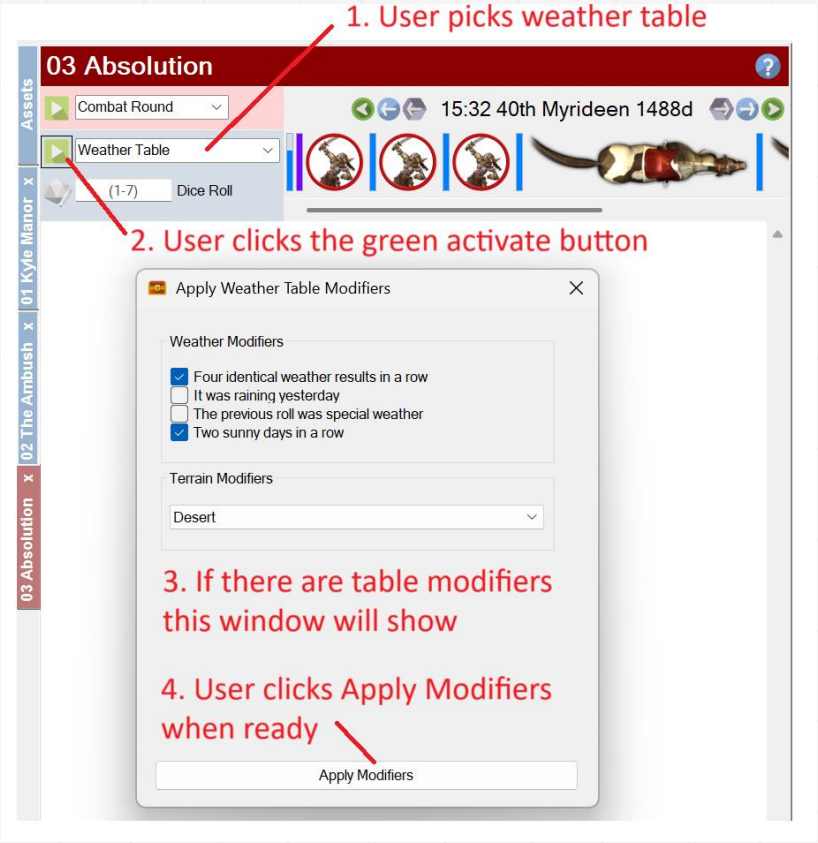


Note that the **Alliances** table is not in the drop down because it is a data table and not a lookup table. Another thing to note is that although the **Weather** table refers to the **Special Weather** table, a user can still choose to roll directly on the **Special Weather** table if they wish.

One other thing to note, is that the two special encounter tables do not appear in the drop down, despite being event tables. This is because both tables have been set to be hidden from the journal drop down:

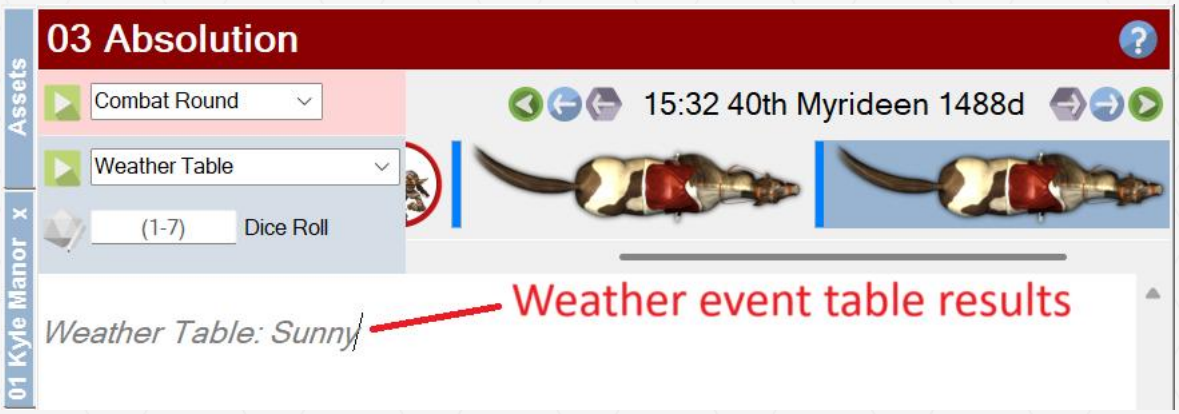


To use a lookup table, a user picks the appropriate table in the journal drop down and then clicks the green button alongside it as shown in the steps below:



Note: The modifiers table will not appear if a lookup table has no modifiers. Instead, the user will be presented with the results as illustrated below. The modifiers table will also appear if the user decides to use the manual dice roll field rather than have Sojour make the roll.

On clicking **Apply Modifiers**, the modifiers are applied to a roll on the main **Weather** table and the journal will be updated as follows:



Had the **Weather** table not had modifiers, the event text would have appeared in the journal right away with no modifiers window being displayed.

Documents

Design History

Before I describe documents in Sojour, I want to provide some design history so that you can better understand why they have evolved in the way that they have.

Sojour was never originally designed to handle documents other than the journals themselves. The reasoning behind this, is that it was intended that all of the mechanical aspects of the game would be handled with physical pen and paper.

Whilst this is how I mostly play, I can appreciate that others might want to go fully digital, so document support got added.

Sojour supports two types of document format: Portable Document Format (PDFs) and Rich Text Format Documents (RTFs).

PDFs were the first natural choice as many role playing game publishers provide digital assets such as character sheets in the PDF format.

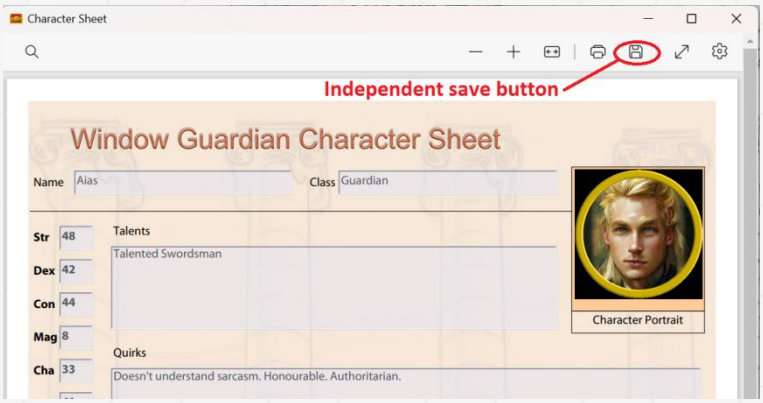
RTF's were added later during testing when it became apparent that some documents may not be available in PDF form for the game system that you wish to play. Whilst Sojour's RTF handling is currently rudimentary, Sojour does allow you to paste in documents in from other sources such as Microsoft Word or perhaps WordPad, which comes with every Microsoft operating system.

This allows one to create a document in a dedicated document editor and take advantage of the full RTF feature set to include such things as tables and images. These can then be pasted into Sojour where the tables and images will remain.

At the time of writing, Sojour is using a free PDF plugin provided by Microsoft. The reason I chose their plugin was solely down to the price. Enterprise level PDF components can run into several thousand pounds GBP.

This left me with the option of providing basic PDF support or none at all. In the end I decided having some PDF support was better than no PDF support.

The upshot of this is that PDFs, such as character sheets, are relatively dumb and that your PDF documents will have to be manually saved every time you change them using the independent save button on the PDF (see below).



Hopefully, if Sojour sells well, I'll be able to upgrade the PDF reader to something with a richer feature set. This will allow for integrated saving and it would also introduce the possibility of being able to programmatically interact with these documents which opens up the flood gates to some really cool functionality.

Document Templates

Sojour has a concept of document templates. A document template represents a singular document that can be copied for use multiple times within a ruleset.

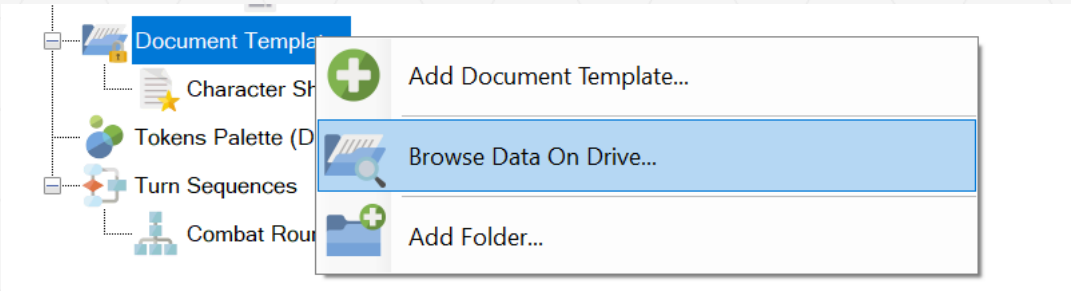
A classic example of a document template is the character sheet. These are documents, but they are always *copied* and filled out to create individual character sheets, one for each character.

Document templates are simply standard PDF or RTF documents that have been chosen to be a template. Once chosen, they become available in a drop down list of documents that a user can choose from when adding new documents to the documents folder.

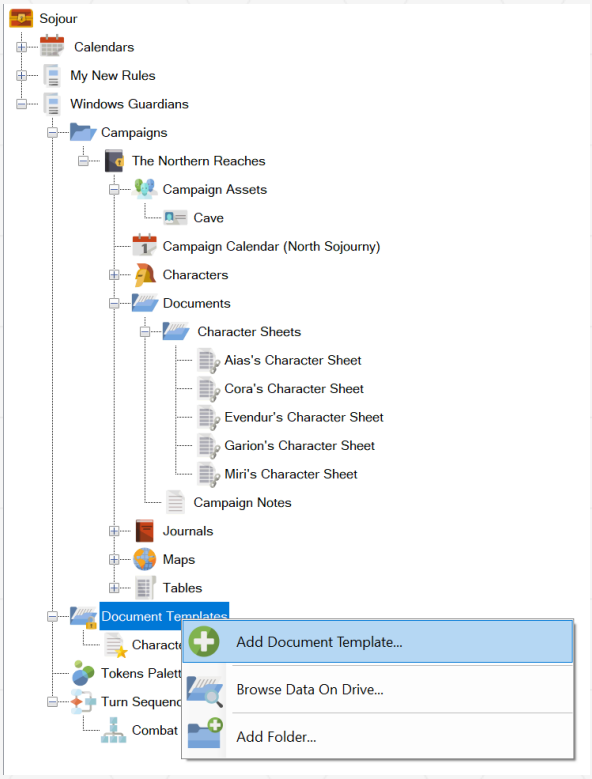
When adding new documents, a user isn't rail-roaded into picking from the list of document templates. They can still choose to add a completely new document if they wish. However, having a list of common documents a mere mouse click away can save valuable time and ensure consistency.

As a further example, consider a ruleset that has a document or form that represents a starship stat sheet. Every time the user creates a new starship they could hunt around their hard-drives for a blank starship stat sheet and then add that. Alternatively, they could import a blank starship stat sheet as a document template. Then whenever they create a new starship, the starship stat sheet will be available in a handy drop down list. This saves time having to hunt around for the stat sheet every time it is needed.

Document Templates are defined at ruleset level. When a document template is created, a copy is made of the original document and is then imported into Sojour. You can view this copy on your hard drive by right clicking the Document Templates folder in the **Assets Browser** and selecting **Browse Data On Drive…**:

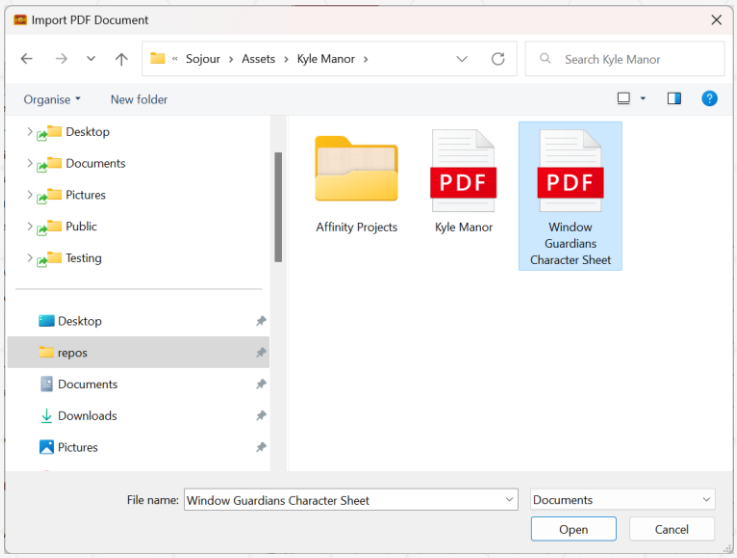


New document templates can be added to Sojour by right clicking the **Document Templates** node in the **Assets Browser** and selecting the **Add Document Template…** menu item:



Alternatively, a document template can also be added by clicking the appropriate main toolbar button when a **Document Templates** folder is highlighted.

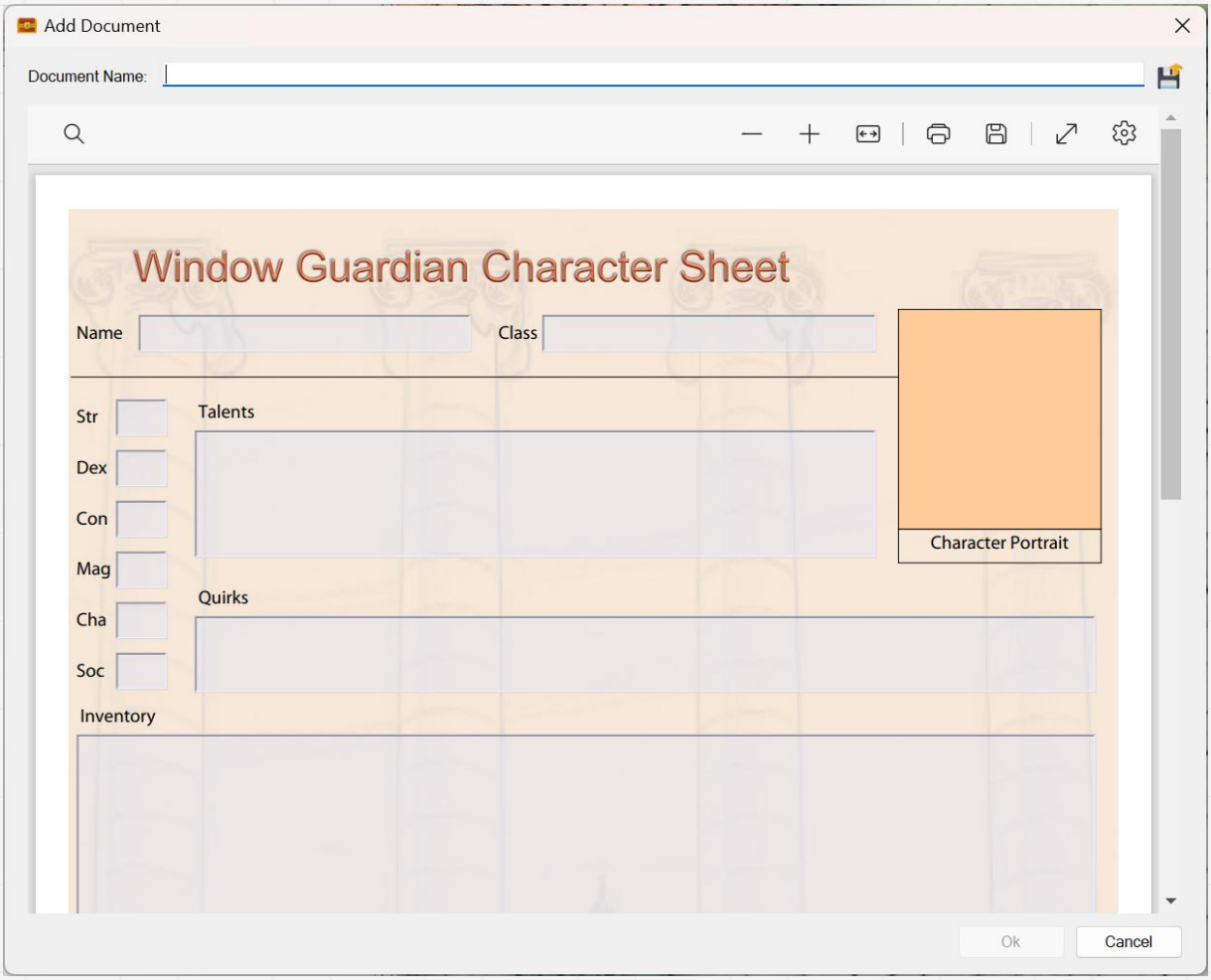
Adding a document template presents the user with a standard windows file browsing dialog to allow the user to select a document as a document template:



The dialog defaults its filtering to show all PDF and RTF document types.

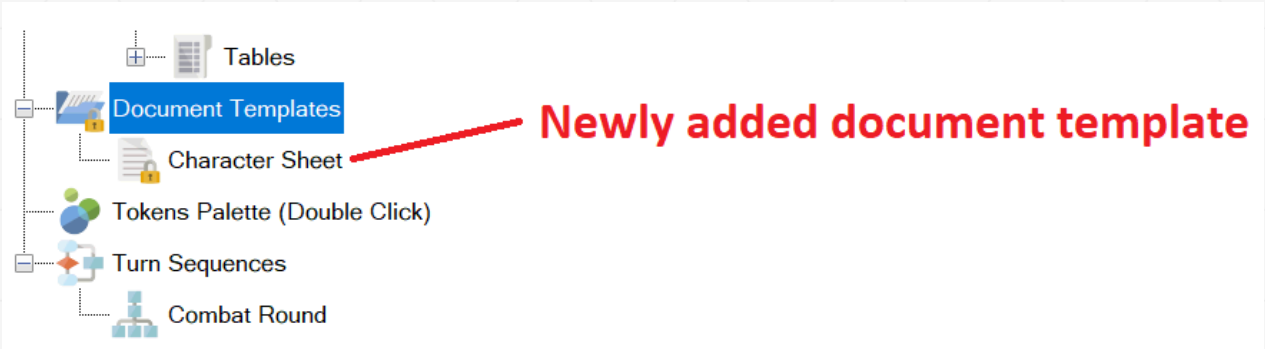
In the example above, the user is about to select a blank Window Guardians character sheet to use as a document template.

Once selected, the user will be presented with this screen:

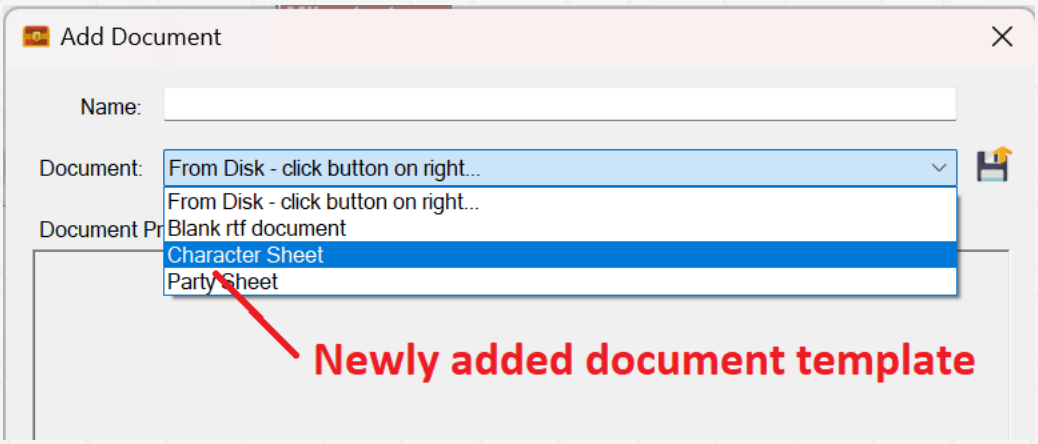


This screen's primary purpose is for the user to provide a document template name. This is the name that will be used to refer to the document template within Sojour. The user can also make basic changes to the document at this stage if it is editable. Don't forget that for PDF's, you have to use the save button on the PDF itself to save PDF changes.

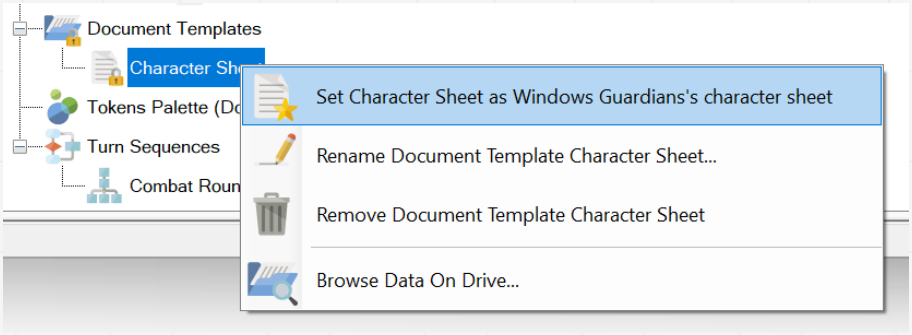
Once named – in this case we called this document template **Character Sheet**, the document will appear under the **Document Templates** folder of the **Assets Browser**:



From here on in, every time a user wants to add a new document to the documents folder, the newly added document template will appear in a drop down list:



Sojour also allows the user to designate any one document template within a ruleset as that ruleset's character sheet. This is achieved by right clicking on any document template and selecting **Set Character Sheet as <ruleset name's> character sheet**:

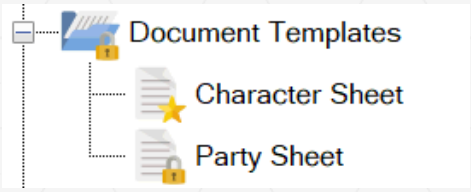


From that point on, any new characters added to that ruleset will automatically get a copy of the selected document template as their character sheet without any kind of user intervention.

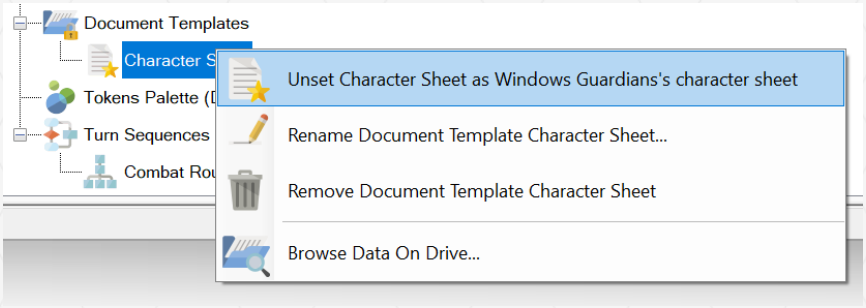
If a document template is designated as a character sheet and another document template is subsequently chosen, the original will transfer its character sheet status to the new document template.

Changing or designating a character sheet will not affect characters that have already been created.

A document template that is set to a character sheet can be easily identified by the yellow star overlaid on it:



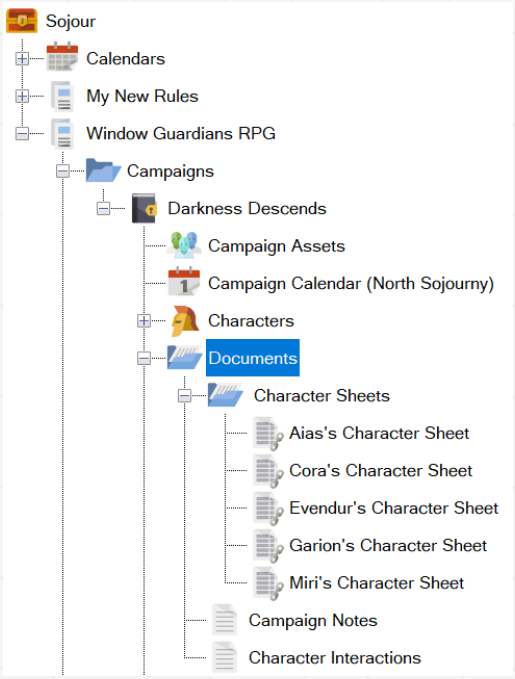
Designated character sheets can be undesignated by right clicking on them and selecting **Unset Character Sheet as <Ruleset name's> character sheet**:



Un-designating a character sheet does not affect existing characters.

Documents

Documents are stored under a campaign's **Documents** folder:



A user can add as many documents as they wish. Documents can be sourced from a document template (it takes a copy), the hard-drive (PDF or RTF's only) or alternatively, a blank editable RTF document can be created.

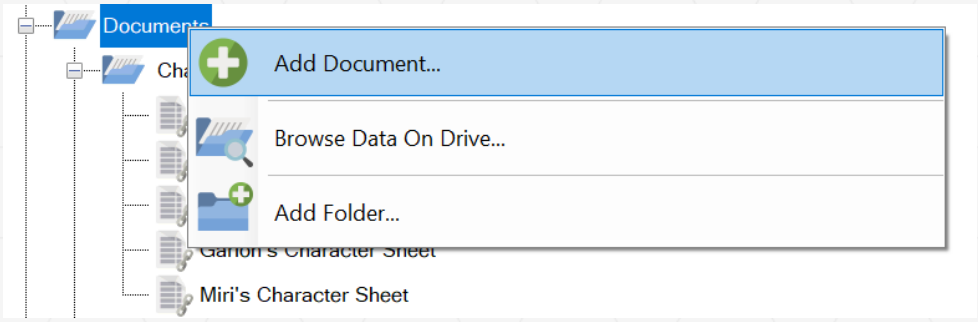
The **Documents** folder has a subfolder within it called **Character Sheets**. The user cannot add documents to this folder. Instead, character sheets are automatically added when a document is associated with a character. See Tokens, characters and campaign assets for more information.

The Character Sheets folder is designed to provide quick access to a character's character sheet and when used, it opens them up as modeless windows. This means that you can open as many as you like and keep working in Sojour.

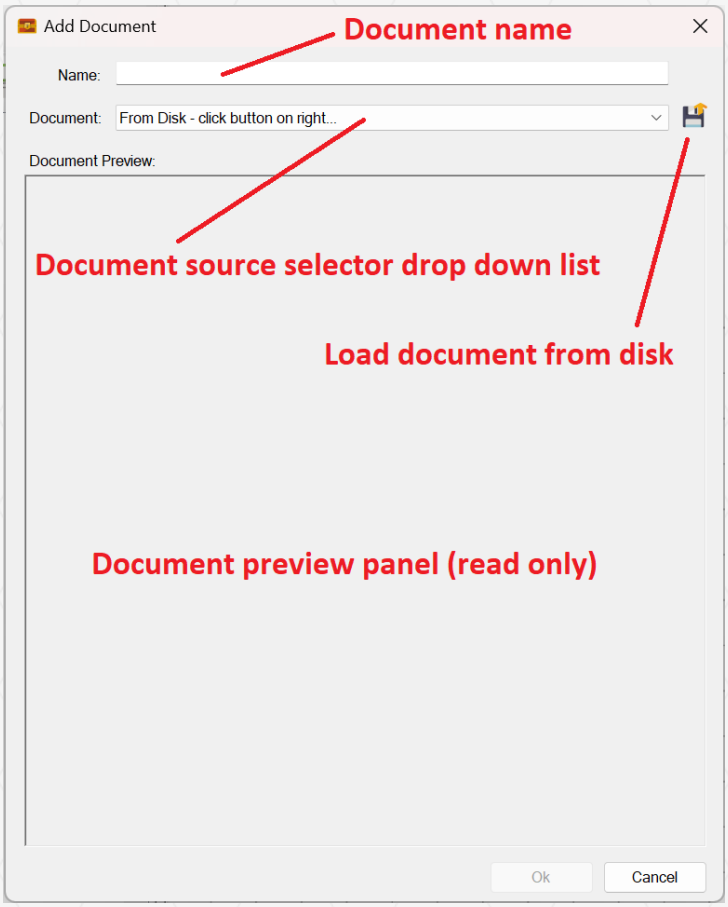
Documents that are added to Sojour have copies made that are then imported. The original document remains untouched and is no longer referred to by Sojour once imported.

Adding a document

To add a new document right click the **Documents** node in the **Assets Browser** and select **Add Document…**:



Doing this will display this window:

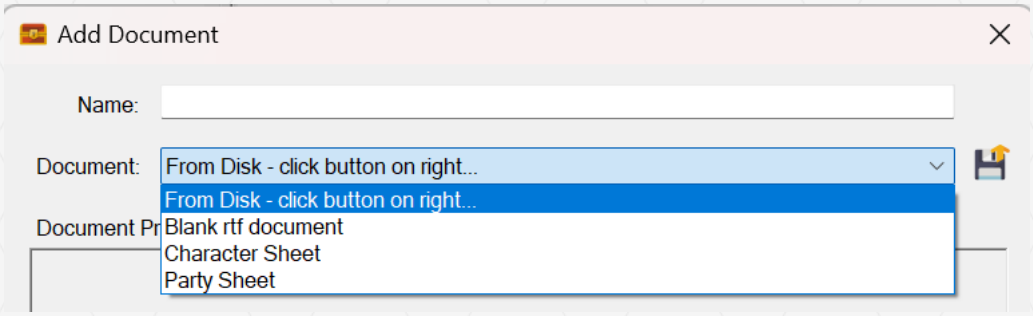


The **Add Document** window allows a user to:

1. Name their document
2. Select their document
3. Preview the selected document

The **Document** field is the most important field as it allows the user to choose how the document is to be obtained.

It has a drop down list with the following options:



1. **From Disk** is selected by default. In this mode, clicking the **Load document from disk** button to the right will allow the user to pick a document from the computer's filing system to be used in Sojour.
2. **Blank rtf document** – Allows the user to create a blank named RTF document that can be edited later.
3. **Document templates** – Below the **Blank rtf document** option is a list of all document templates registered with the ruleset. In the above example this ruleset has a **Character Sheet** and a **Vehicle Sheet** registered as document templates. If the user selects a document template, Sojour will make a copy of it and import it into the campaign as a document with the name given in the name field.

Once a document is chosen, all the user need do is name it in the **Name** field, then Sojour will import the document into the campaign. The newly imported document will appear under that campaign's **Document** folder in the **Assets Browser**.

Documents in Sojour are either PDFs or RTFs. Sojour tries to treat both in the same way, but there are limitations due to the limited functionality of the PDF reader being used.

Sojour will remember the document zoom level, window position and size of a document's window (all Sojour document windows can be resized).

The next time the document is re-opened it will reappear in the same location on the desktop where it was last used, with a window set to the same size and with the same zoom setting as last used.

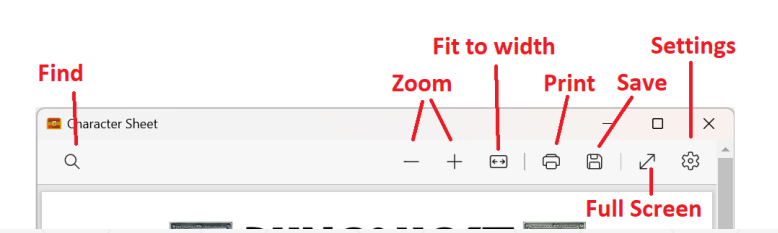
All Sojour document windows are modeless. This means that you can open as many of them as you like and you can continue to operate Sojour as if they aren't there.

Warning! Sojour's map pane captures all keyboard input whenever the mouse is hovering directly over the map. Where this occurs, no keyboard input will be forwarded to an open document. To rectify, simply move the mouse cursor away from the map, or over a document, even if that document is over the map pane.

The specifics of PDF and RTF documents are discussed below:

PDF Documents

PDF documents use a third party PDF reader. These can be edited if the source PDF is editable. However, as discussed in the document Design History section, the only way to save these documents is with the save button on the document page. Sojour will *not* automatically save PDF documents.



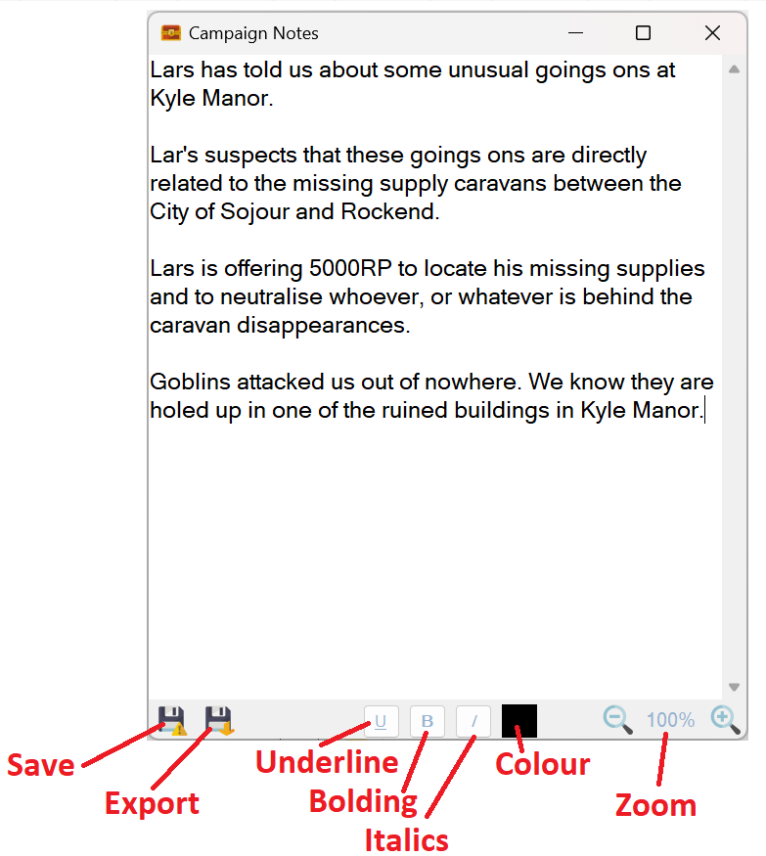
The functions are as follows:

1. **Find** – Finds text within the document.
2. **Zoom** – Alters the zoom level of the document.
3. **Fit to width** – Automatically sizes the document to fit the width of the window. All document windows are resizable.
4. **Print** – Print the document using the Windows print system.
5. **Save** – Saves your document. As noted above, you must use this to save your changes. Sojour cannot save these PDF documents as the third party API doesn't provide access to this function.
6. **Full Screen** – Puts the PDF into full-screen mode. Testing has shown this functionality to be problematic. I recommend that it shouldn't be used. It's a feature of the third party plugin that doesn't appear to be fully functional.
7. **Settings** – Allows you to view and alter various PDF related settings.

Note that at the time of writing the plug in does not fully support PDF image fields. It will display them, but it won't allow you to add an image to it. Adding an image will need to be done in an external PDF reader.

RTF Documents

A typical opened RTF document is shown below:



RTF documents use the same component as Sojour's journals, as such, they are provided with a number of common features:

1. **Spell checking** – Badly spelled words are temporarily highlighted in red and the user can right click on them to check the spelling. More information can be found here.
2. **Automatic save** – RTF documents are always saved automatically when closed. An option is also provided to allow the user to manually save. More information can be found here.
3. **Export to an external RTF** – Allows the document to be exported as an external RTF document. More information can be found here.
4. **Font changes** – Simple changes to font such as colour and style. More information can be found here.
5. **Built in dice rollers** – Using the [] and {} brackets allows the user to type in dice expressions that will be automatically resolved within the document. More information can be found here.
6. **Document Zoom** – The document can be zoomed in or out and the document will remember its last zoom setting. More information can be found here.

Tokens, characters and campaign assets

Sojour supports a wide variety of map token types that each utilise the same internal code. However, each of the token types is designed for a different use.

We are going to use the word token to represent a map token that could be associated with tokens, characters or campaign assets.

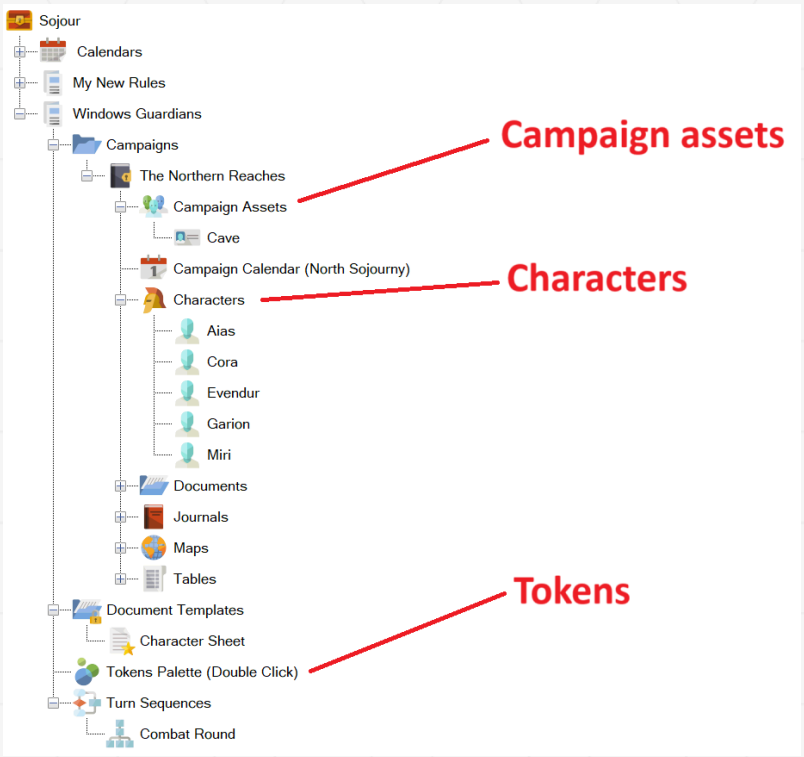
In essence, all token types consist of a map token image that is paired with specific data and an optional document. The optional document can be any PDF or RTF document, but in general they tend to be some kind of stat or character sheet.

Token Types

Tokens have three main types in Sojour:

1. **Campaign Asset** – These represent unique campaign assets that can be shown on the map. A good example of a campaign asset might be a starship in a Sci-Fi campaign.
2. **Characters** – These are the main characters within a campaign. Sojour makes no distinction between party characters and other prominent campaign characters – both will appear here. Characters are unique and will appear on the character bar when active.
3. **Tokens** – These represent your monsters or other mobs within the campaign. Tokens are not unique and can be instanced many times. For example, you might have a single Goblin token, but this token can be dragged to a map multiple times to create many Goblins.

All token types are accessible from the **Assets Browser**:

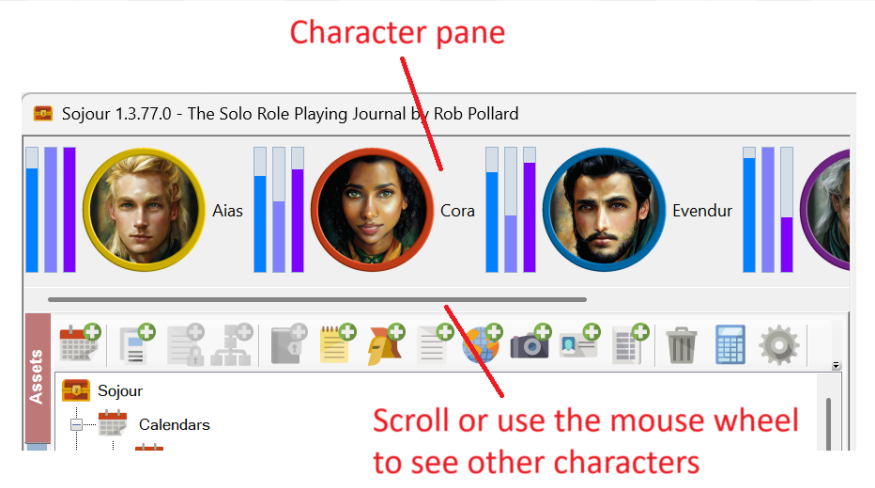


Characters and campaign assets are defined at campaign level, whilst tokens are defined at ruleset level. This can be confirmed by examining the **Assets Browser** screenshot above.

The table below summarizes the differences between the three token types:

Token Type	Level	Number of map instances	Character Pane?	Character sheet document templates	Can be deactivated?	Journal npc bar?	Can be made static?	Hide map tooltips?
Campaign asset	Campaign	Unique or Multiple	N	N	N	Y/N	Y	Y
Characters	Campaign	1 (Unique)	Y	Y	Y	N	N	N
Tokens	Ruleset	Multiple	N	N	N	Y	N	N

- **Level** – Determines where the token is defined. For example, if a token is defined at campaign level it would be unique to that campaign.
- **Number of map instances** – Determines how many copies of the token that you can create on a single map. Campaign assets can be toggled between unique or multiple.
- **Character pane?** – Determines if the token is visible on the main character pane/toolbar. The character pane shows characters in alphabetical order. The character pane is shown below:

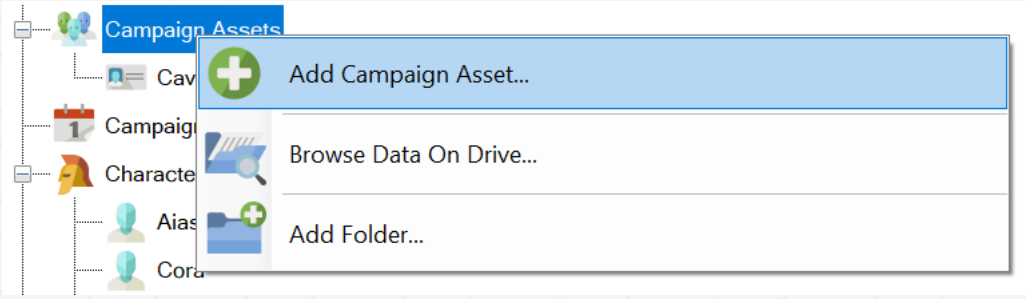


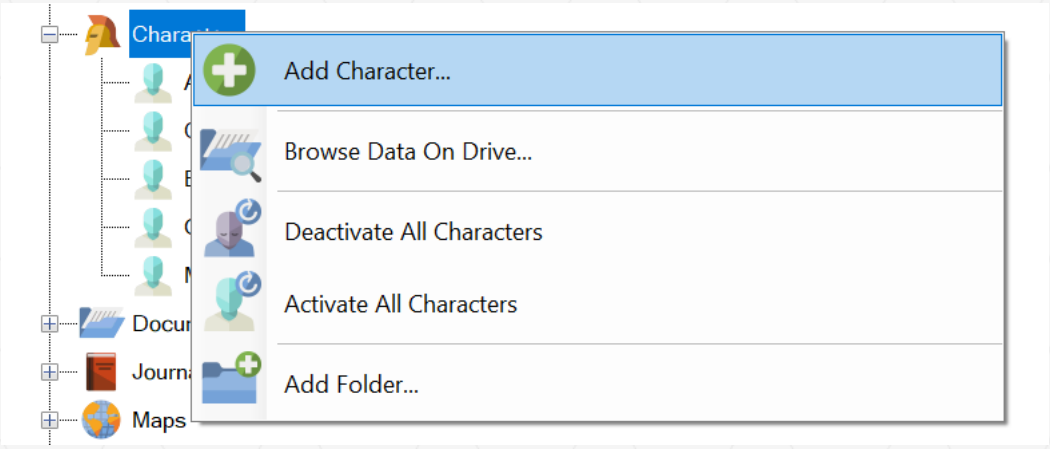
- **Character sheet document templates** – These tokens can have a dedicated document template defined as the character sheet for that ruleset. Every time a new token of this type is created, it automatically gets a copy of that character sheet associated with it.
- **Can be deactivated** – Character type tokens can be activated or deactivated dependent on whether or not you want to make them available for immediate use.
- **Journal NPC bar**? – All token typed tokens are treated as npcs. Campaign assets that have characteristics will also be shown here too. They will always appear in a journal's npc bar when they are present on the current opened map:

- **Can be made static?** – Allows the user to decide if the token is fixed to the map and cannot be moved. Handy for providing new location graphics on the map as new things are discovered.
- **Hide map tooltips?** – Allows a user to specify if the token shows any tooltips when the mouse is waved over it on the map.

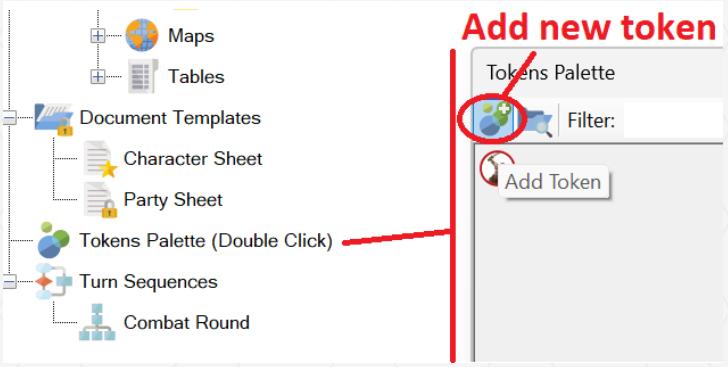
Adding tokens, characters and campaign assets

New campaign assets and characters can be added by right clicking the appropriate **Assets Browser** node and selecting the **Add XXXXXX** option on the context menu:



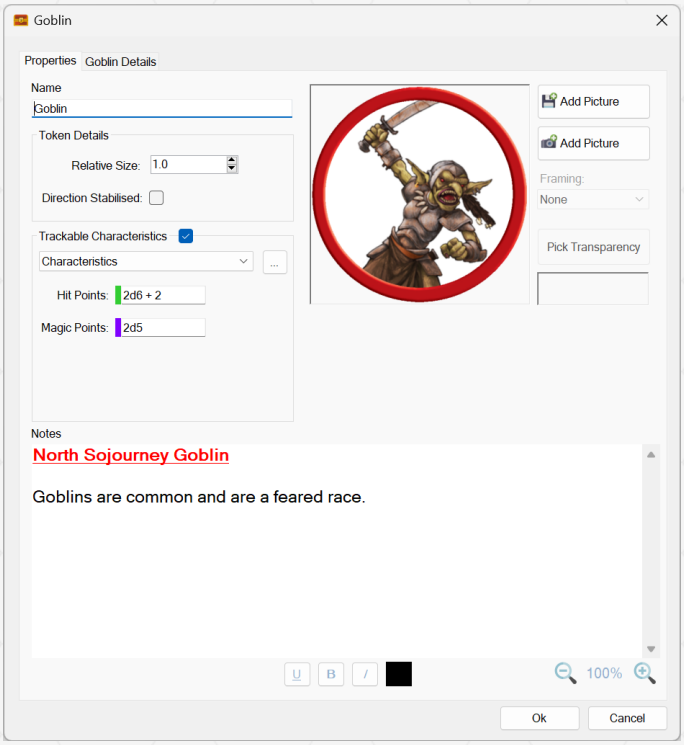


Tokens are a little different in that you must first double click the **Tokens Palette** node to open the token palette and then click the **Add Token** button:



Token details

Irrespective of the type of token you choose to add, you will be presented with a window that looks similar to the one below:



Each token is defined by a number of fields which are described below:

Name

The name of the character, campaign asset or token.

Token typed tokens can be dragged to the map multiple times to create many copies. Each copy will have a number appended to its name. For example, if we were to drag this Goblin token to the map three times, we would end up with three different Goblins named Goblin 1, Goblin 2 and Goblin 3.

Dragging characters or campaign assets to a map several times will simply result in the same token being moved on the map.

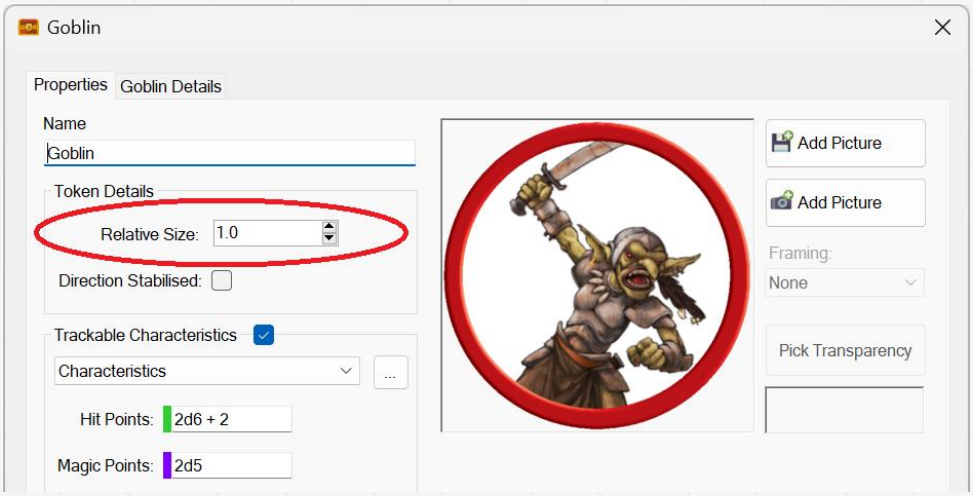
Changing the name will alter the name of that token on all maps.

Relative Size

Unlike other VTTs, Sojour unifies the scale of all token images. All tokens will appear at exactly the same size on the map and character pane, irrespective of their source image's original size.

This allows the user to quickly capture token images from a variety of sources without needing to worry about scaling them all to be the same size.

However, there will be occasions where a user might want to vary the size of a token on the map. A typical example might be for a Dragon. On the whole you want those to be big! To support varying sizes, Sojour provides a **Relative Size** setting as shown below:



Relative size makes the longest side of an image equal to n metres in length where n is the relative size. Or to put it another way, Sojour is designed so that humans are size 1.0 and their tokens will appear as roughly 1 metre in radius on most maps**.

** Maps will intelligently increase the token sizes on large maps to make them visible. If it didn't do this then the tokens would be very difficult to see where the map sides are many kilometres/miles in length.

When choosing a relative size for your token, think about its relative size to a human and simply set the relative size appropriately. As noted above, this relative size is independent of the source image size.

Relative size also accepts fractional values smaller than 1.0 to allow the user to create tokens for creatures or items that are smaller than a human.

Regardless of a token's chosen relative size, a user is always free to change the size of a token on the map using <Ctrl> mouse wheel over the token. This overrides the size of that single instance of the token and will persist between sessions.

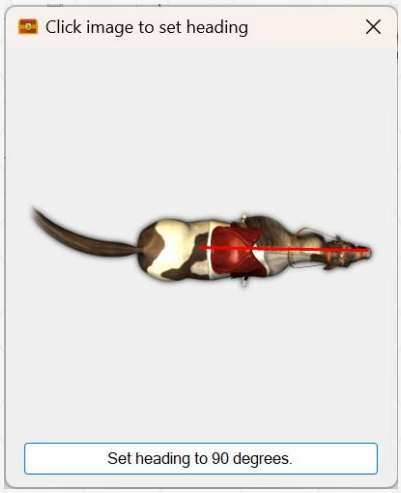
Direction Stabilised

This tick box determines whether a token is direction stabilised on the map. **Direction stabilised** tokens always face their direction of movement. Non direction stabilised tokens always face the same way.

Direction stabilised defaults to off.

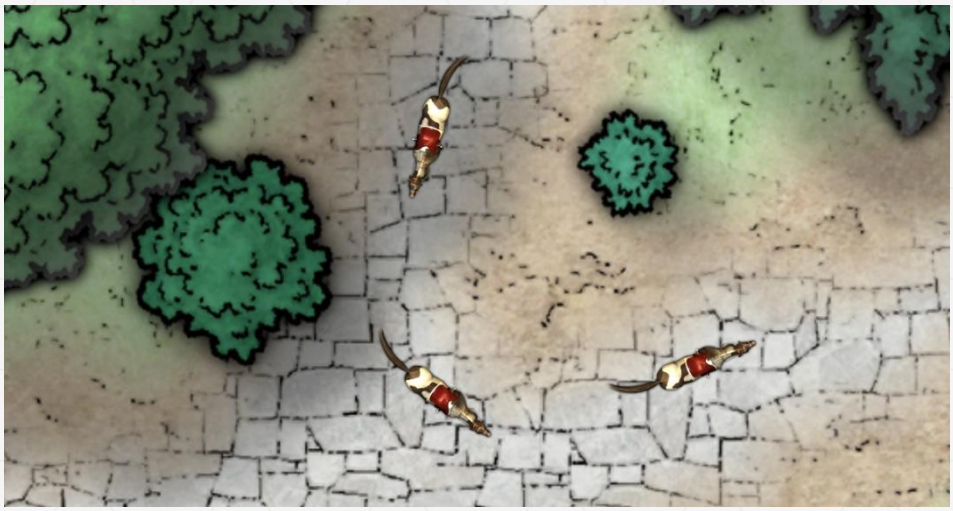
Irrespective of whether a token is **Direction stabilised** or not, it can be manually rotated on the map by using the mouse wheel directly over a token.

Direction stabilised can only be selected once an image has been picked using either of the two **Add Picture** buttons. This is necessary as the user must pick the 'front' of the token on the window below when **Direction stabilised** is enabled:



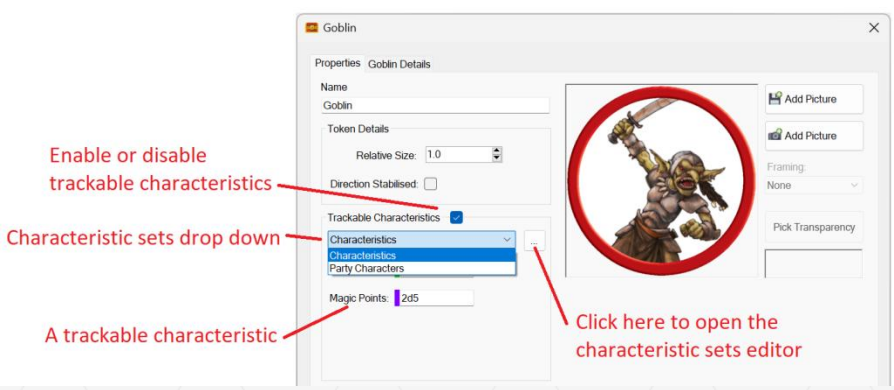
The front is chosen by left clicking on the image. Here we have clicked the horse's nose resulting in the token's front being set to a heading of 90 degrees.

When **Direction stabilised** tokens are dragged to the map, the token's front will always face the direction of travel:



Trackable Characteristics

Every entity (campaign asset, character or token), has the option to enable trackable characteristics. These consist from zero to four user made fields. Each trackable characteristic is part of a characteristics set. The drop down allows you to pick a characteristic set that is appropriate to the entity you are editing:

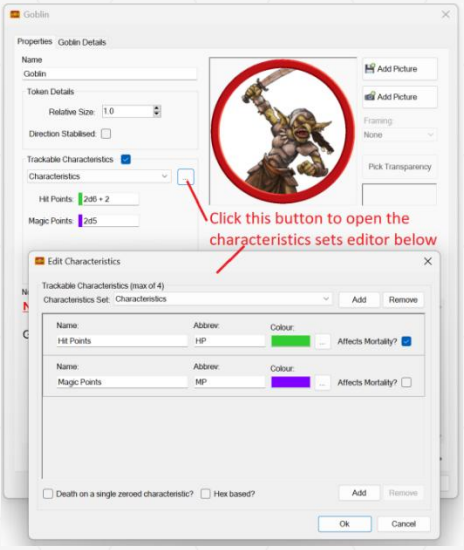


Once a characteristic set is selected, the appropriate trackable characteristics for that set will be displayed.

Warning! All trackable characteristic controls will be disabled if **Trackable Characteristics** is disabled.

To disable a specific trackable characteristic field for the entity, enter 0 (zero) into its field. This is handy as it allows you to pick a subset of the available fields.

If you need additional fields, or perhaps a whole new set of trackable characteristics, click the **…** button to open the characteristic sets editor:



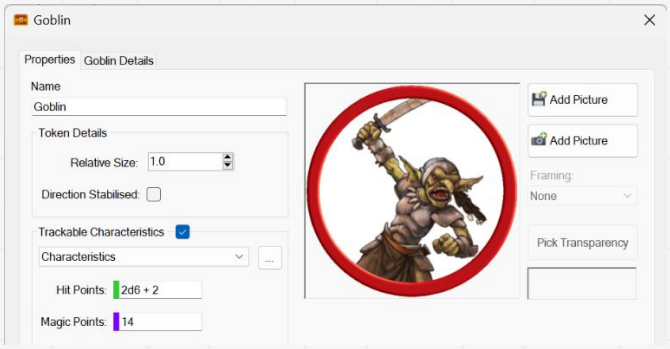
This editor will allow you to add or edit existing fields or create whole new characteristic sets. More details can be found here.

If trackable characteristics are enabled, you will not be able to click the Ok button on the dialog window until all the trackable characteristic fields have valid values in them.

Character type entities and campaign assets set as unique can only accept numerical values into these fields, with a 0 (zero) disabling that field. If the chosen characteristic set was set up as a hexadecimal set, then hexadecimal numbers are also valid in this field.

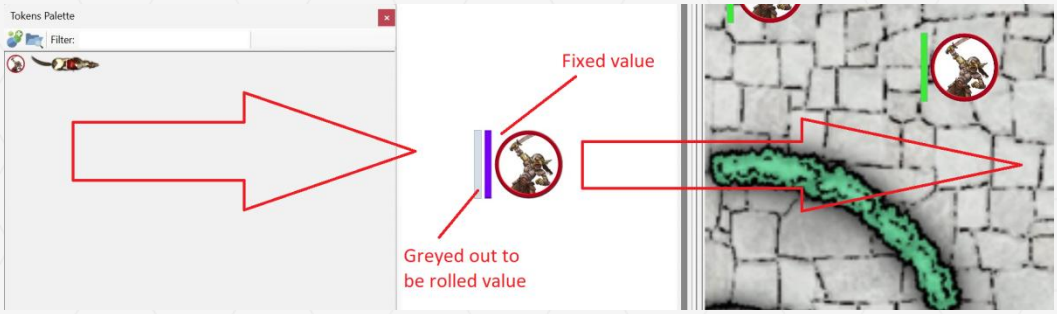
The numeric value entered represents the maximum points that entity has in that characteristic. This sets the maximum level that the characteristic can be set to from its health-bars.

Token type entities and campaign assets set as multiple instanced can also accept dice expressions in these fields as shown below:

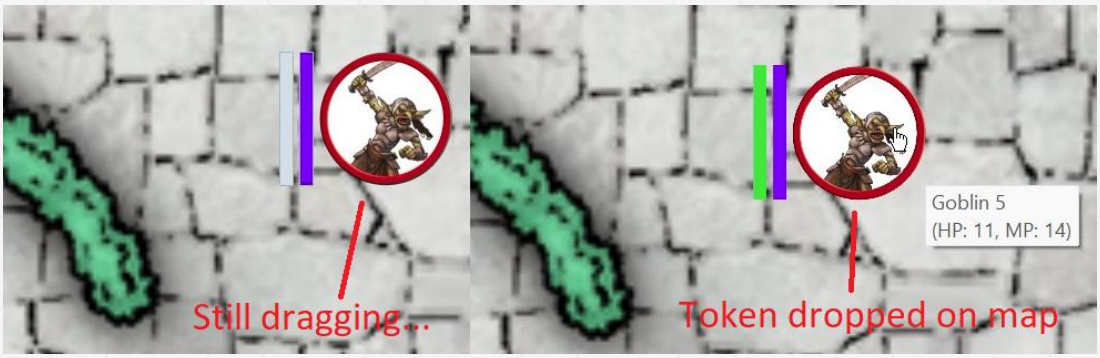


Where a dice expression is present, a new individual characteristic maximum will be rolled each time that token or campaign asset is dragged to the map.

For example, if the above goblin was dragged to the map:



One would see that the **Magic Points** bar is coloured because it already has a value of **14**. However, the **Hit Points** health bar being greyed out – as this currently has no value until the token is dragged and dropped onto the map. Once dropped, a 2d6 + 2 will be rolled for and the bar will change colour to show that it now has a value:

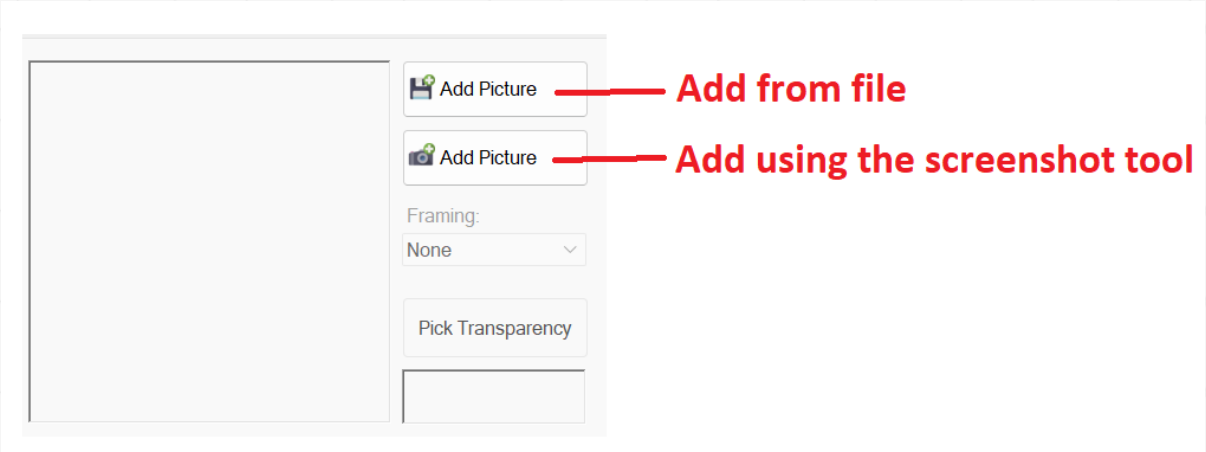


Notes

The notes field allows the user to add notes specific to the token.

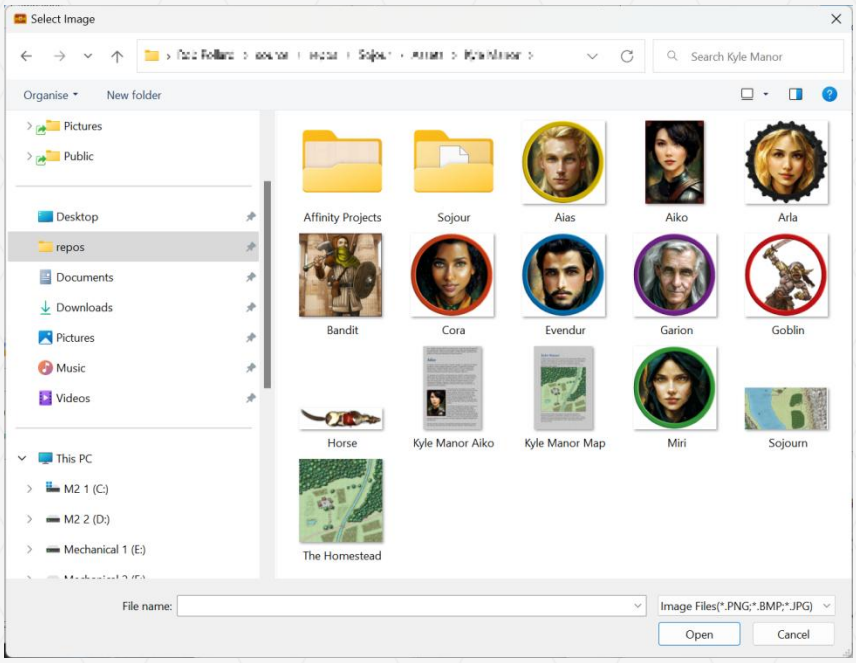
Token Pictures

Two buttons are provided for adding a picture for a token:



Adding a token from an image

The topmost button opens a standard file browsing window to enable the user to pick an image from their filing system:



Sojour currently supports BMP, PNG and JPG file formats for tokens.

All imported token images are auto-scaled and default to a map size where its largest dimension is 1 metre in length. This scaling is irrespective of the token image's original pixel size. For more information read the Relative Size section.

Adding a token from a screenshot

The alternative way of adding a token to Sojour is to use the built in screenshot tool. This is a great tool for grabbing images directly from your game's digital documents.

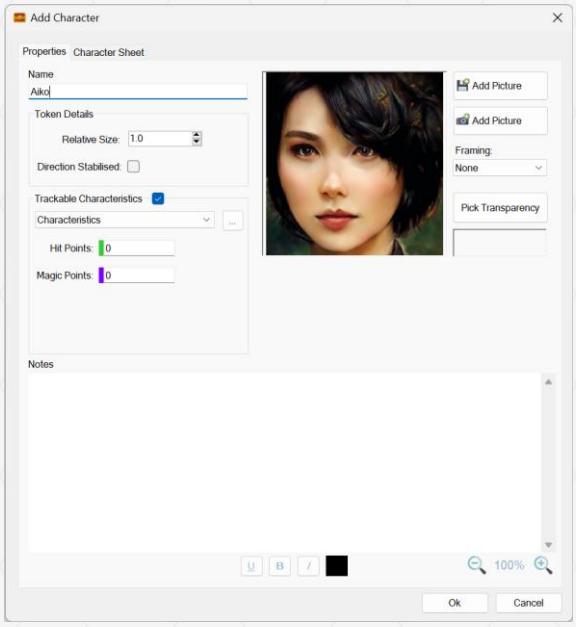
Before using this tool, first ensure that the image is already visible on your desktop. The tool automatically hides Sojour and all of its windows when taking the screenshot so there is no need to worry about moving Sojour out of the way.

The tool looks like this in operation:

The first left click defines the top left of the capture window. The second left click determines the bottom right of the capture window.

Unlike the Windows screen capture utility, Sojour does not require that the user keep the left mouse button held down whilst sizing the capture area. All that's needed is a second left click to define the bottom right corner.

Regardless of which method is used to import the token image, you will end up with the imported image displayed in the **Add Character** window as follows:

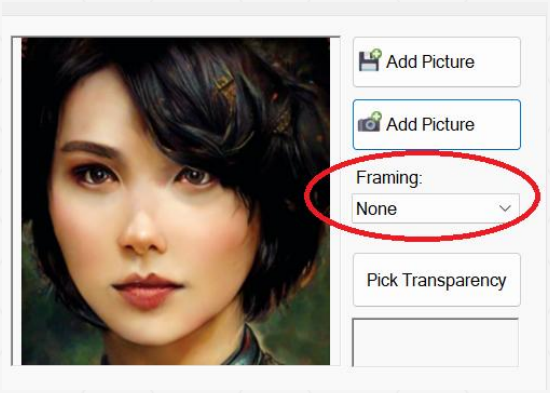


Token Framing

Whilst the example imported image could be used as-is, it wouldn't look very good on the map:

Sojour provides a number of tools to help clean up an imported token image.

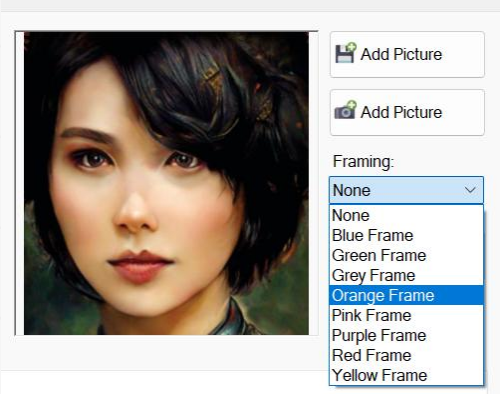
The first tool we will look at is **Framing**:



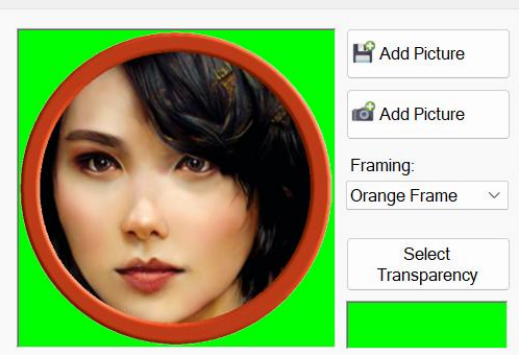
Framing converts any token picture into a standard round character token complete with a coloured frame and transparency.

Warning! Framing can only be used during the initial import of the picture. It cannot subsequently be used and will be disabled. If you want to frame your token's image, you must do so immediately after the picture is captured.

To frame a picture simply pick one of the framing options from the **Framing** drop down list:



In the above screenshot we will pick an orange frame. After picking the orange frame the picture window will look like this:



Sojour has automatically created a round orange frame and has set a transparency area that is shown in green.

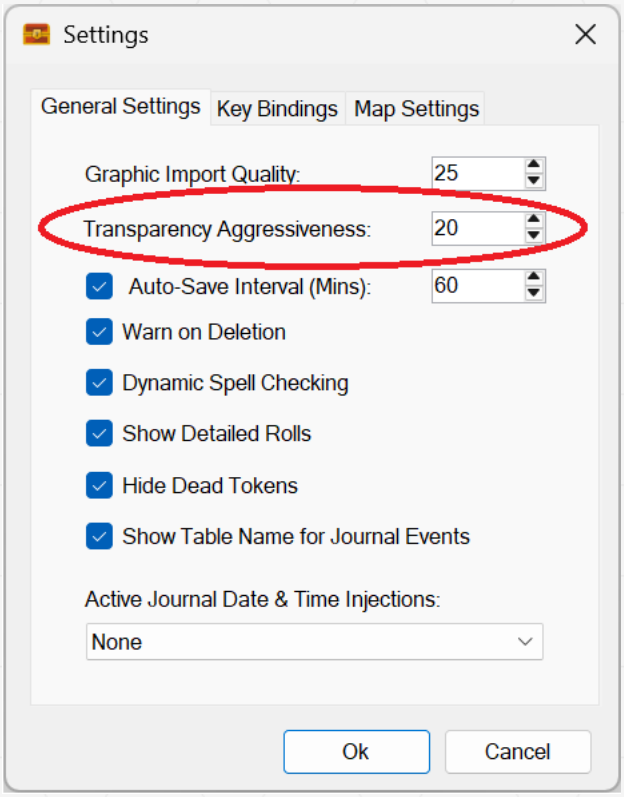
The token will now look like this when dragged to the map:

This is an obvious improvement over the original image.

Token Transparency

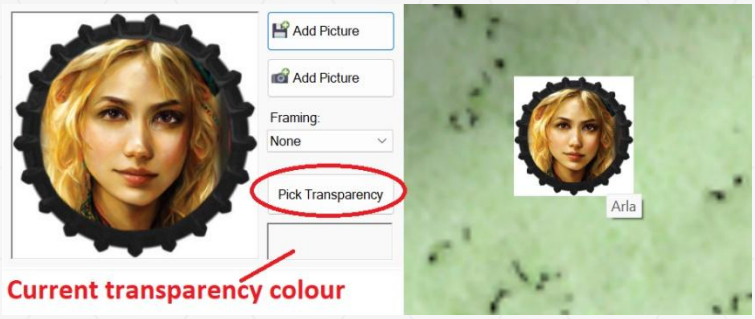
In addition to framing, Sojour also allows the user to select a specific image colour to be set as transparent. Sojour will pick that colour and all similar colours and set them all to transparent.

The aggressiveness of this colour selection is a global setting within Sojour's settings window:



Higher values are more aggressive.

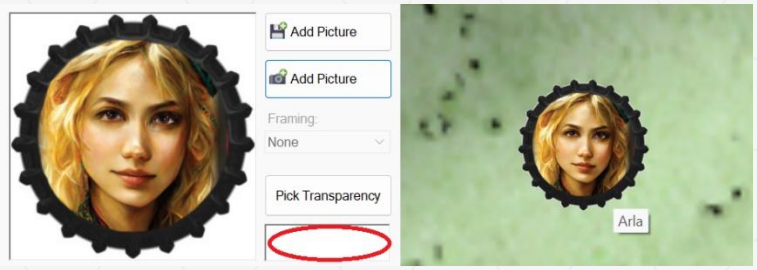
To set the transparency for a token's image, click the **Pick Transparency** button and then click anywhere on the image to set that colour to transparent:



In this example, Arla's background is white, this can be most easily seen when dragging Arla's token to the map as shown on the right.

It can also be seen a little more subtly within the window shown on the left. The transparency colour located just below the **Pick Transparency** button is a slightly different colour from the white surrounding the Arla's token (grey rather than white).

To fix Arla's token's transparency we would click the **Pick Transparency** button, then we would left click anywhere in the white area outside the token. This will result in the transparency colour being set to white as confirmed by the transparency colour box on the left:



Dragging Arla to the map now results in a much more professional looking token as the white surrounds have been set to transparent.

Warning! The **Pick Transparency** function will set all colours and similar colours located *anywhere* on the image to transparent.

Warning! **Pick Transparency** can only be used during the initial import of the picture. It cannot subsequently be used and will be disabled. If you want to set your token's transparency, you must do so immediately after the picture is captured.

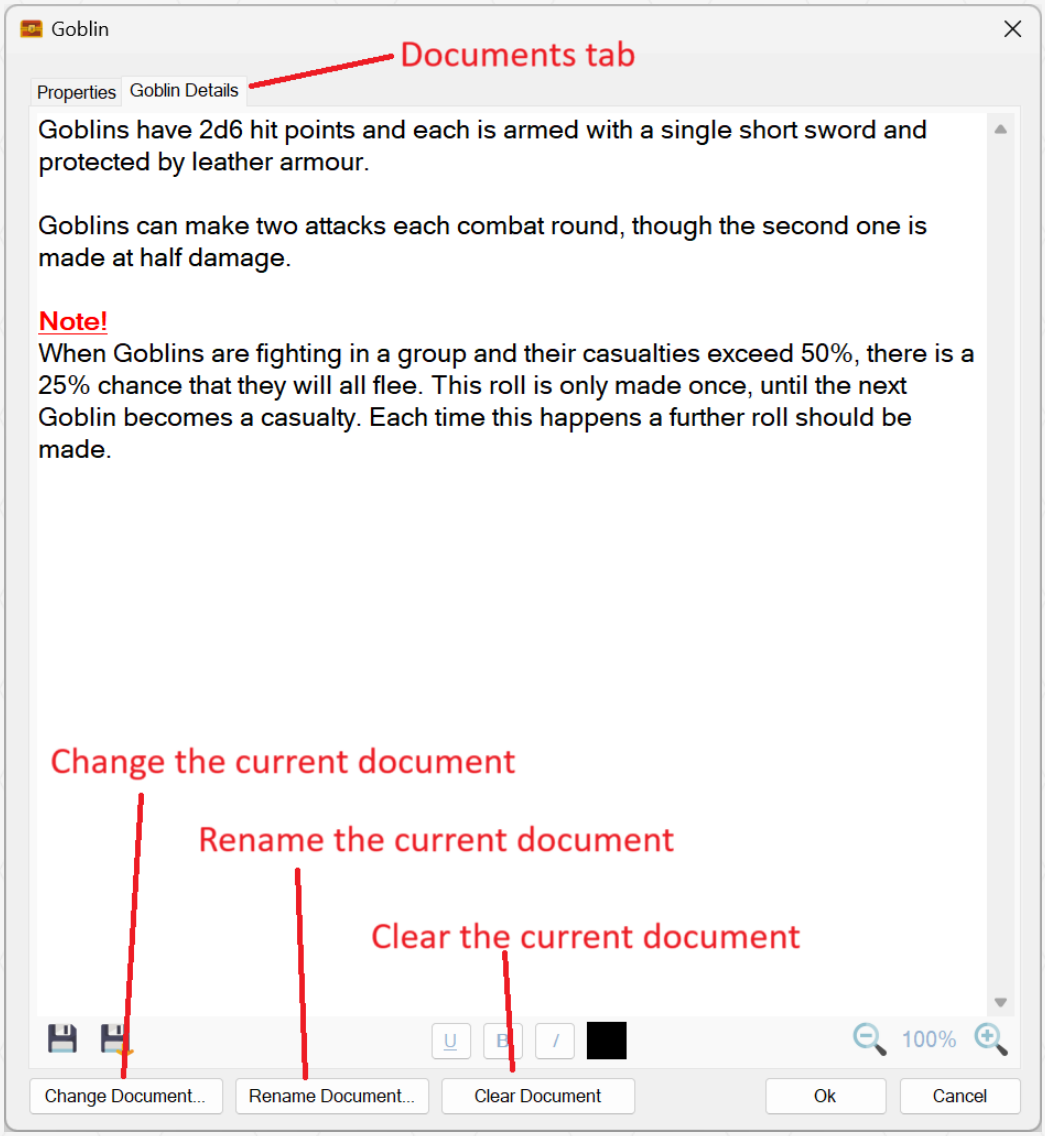
If there are other areas on the image that you don't want to make transparent that have the same colour as the area you are targeting, I highly recommend that you change the colour of the area to be made transparent into a colour that does not appear anywhere else on the image. This will ensure that the transparency algorithm only targets the parts of the image that you want it to.

This colouring will need to be performed in an external art package like windows Paint.

The combination of transparency setting, framing and the screenshot tool should allow you to easily create token images directly from any digital source!

Token Documents

All token types can have up to one document associated with them. This document, like all documents in Sojour must be an RTF or a PDF. A token's document will appear in the document tab – the rightmost tab:



The document tab's name is the same as the document that it holds. In this case, we created a RTF document for the Goblin and called it **Goblin Stats** – but you could call it anything. Characters tend to have character sheet PDFs associated with them, so this tab will more than likely be named 'Character Sheet'.

The user can change the current document for a new one using the **Change Document…** button.

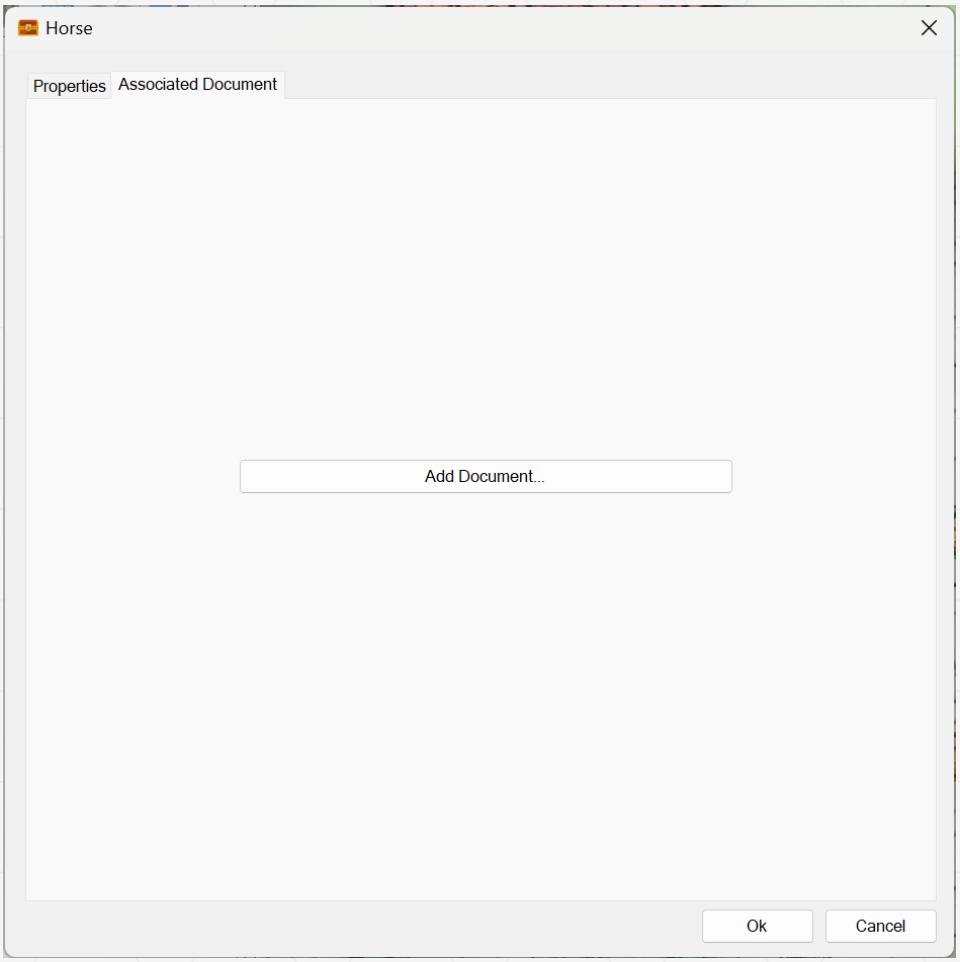
In addition they can rename the document by clicking the **Rename Document…** button.

The final option, **Clear Document** allows a user to remove and clear the currently attached document.

A token's documents can be opened on the map by either double clicking the token or by right clicking and selecting **Open <Document Name>…**

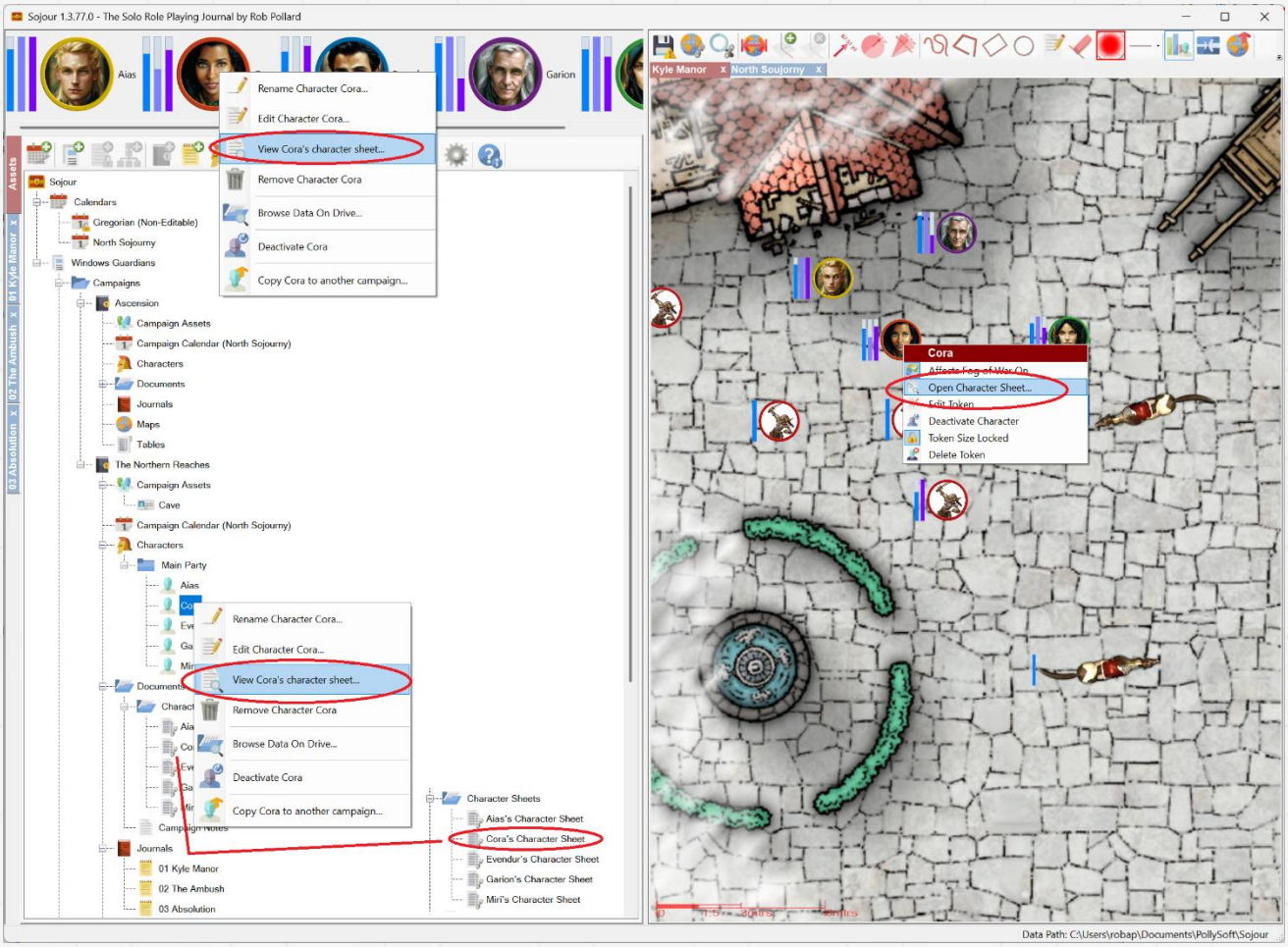
For more information on documents please review the documents section of this manual.

In general, token types are not given a document by default. In this case, their documents tab will be named **Associated Document** will look like this:



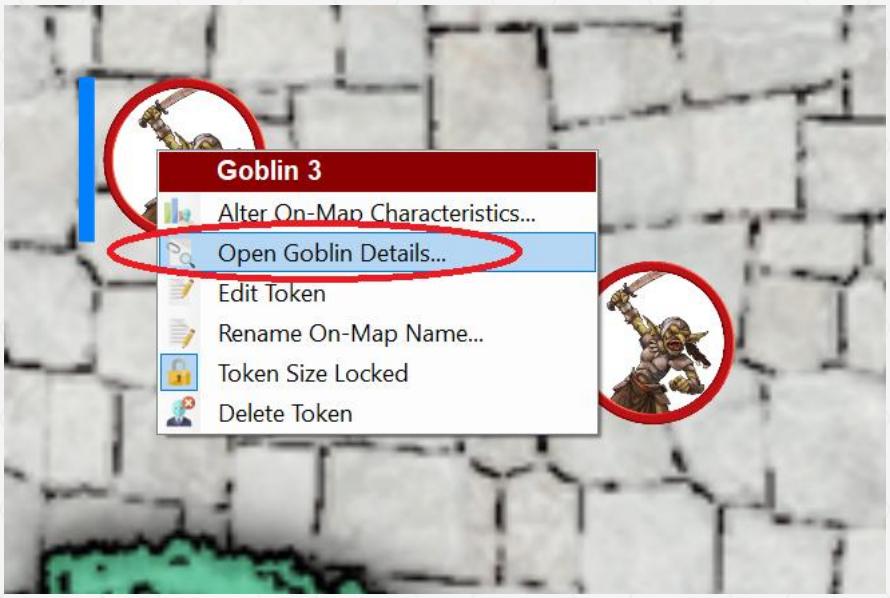
To add a document simply click the **Add Document** button in the centre of the window. Adding a document presents you with the standard Sojour Add Document window.

A character's document can be viewed by right clicking the token on the map, or its node in the **Assets Browser** or its character image on the character pane or by opening that character's character sheet in the **Assets Browser**:



The language above uses the phrase ….*Character Sheet*, however, this will change to reflect what the associated document is actually called.

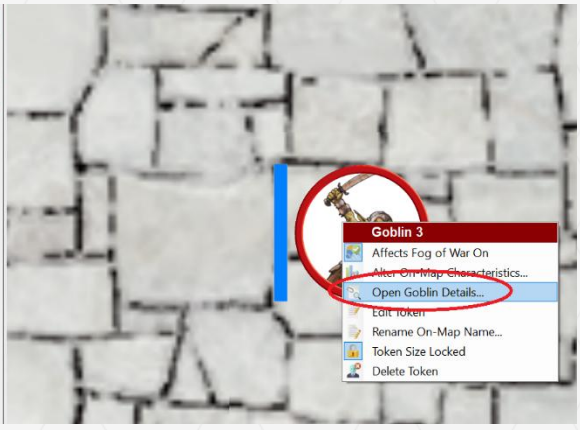
For example, a Goblin npc's associated document might be called **Goblin Details**, in which case it will appear as **Goblin Details** in the various menus:



If a character has no associated document, the menu option to view it will not be visible.

Campaign assets can have their associated document viewed from the map or by opening the campaign asset from within the **Assets Browser**.

Tokens can have their associated document viewed from the map, the journal's npc bar or by right clicking the token in the token's palette:



Note that in this case, the Goblin's associated document was called **Goblin Stats**.

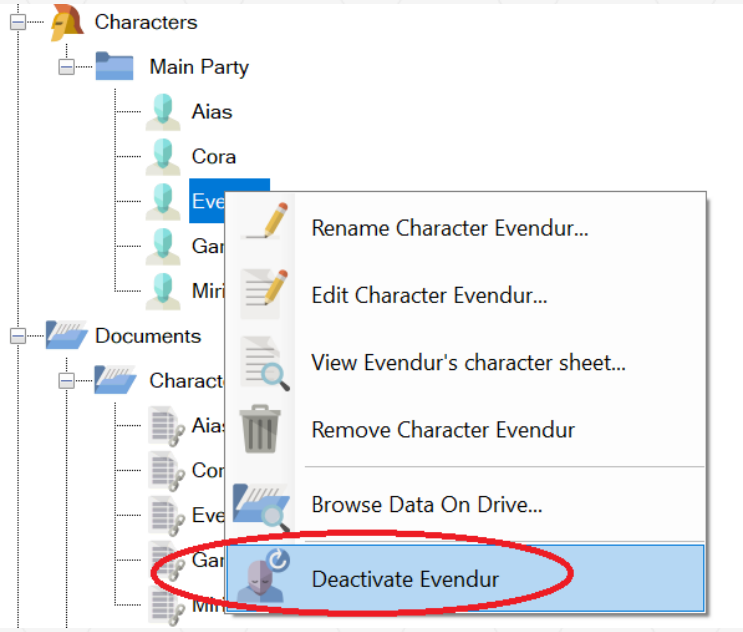
Additional character functionality

Characters have additional functionality that isn't available for campaign assets or tokens. This additional functionality is described below.

Activating and deactivating characters

Characters can be activated or deactivated as needed. This functionality is useful for campaigns where many characters come and go over time.

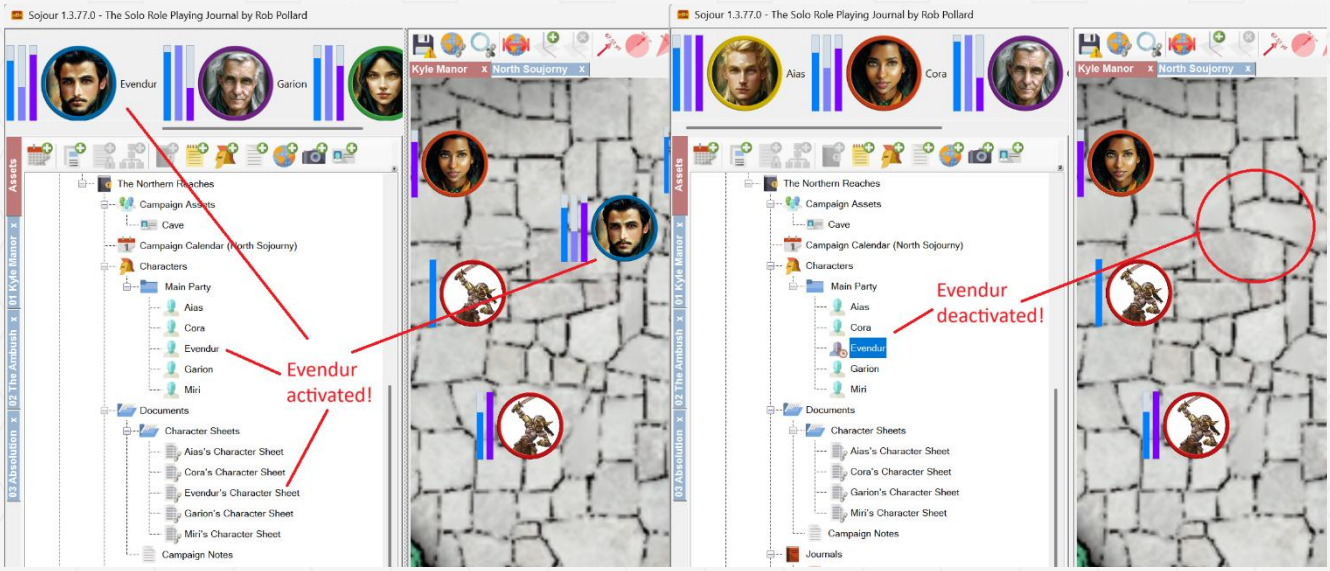
A character can be deactivated by right clicking them and selecting **Deactivate <<Character Name>>**:



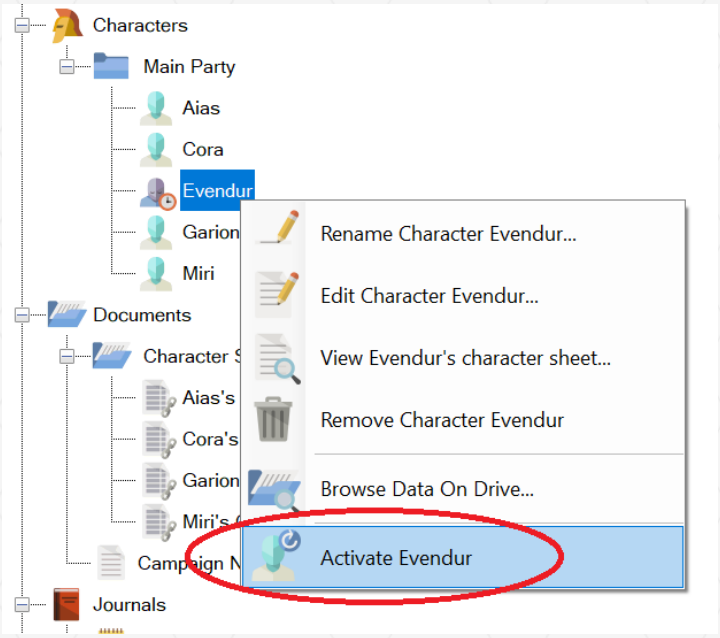
Deactivating a character results in the following happening:

1. The character's icon in the assets browser will switch to being asleep with a clock icon  .
2. The character is removed from the main character toolbar.
3. The character's character sheets are removed from the **Character Sheets** folder under **Documents**.
4. The character is removed from all maps – though Sojour does remember where they were, so they will qwawaqasbe restored to the same places when they are reactivated.

The screenshot below shows a before and after for the deactivation of **Evendur**:



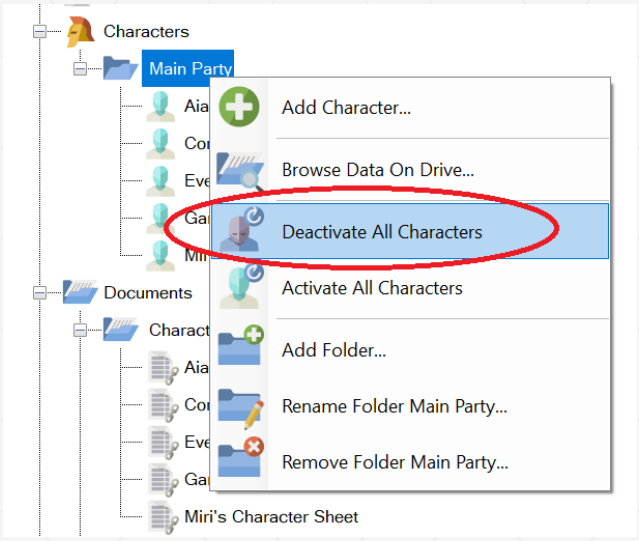
To reactivate a character, right click it and select **Activate <<Character Name>>**:



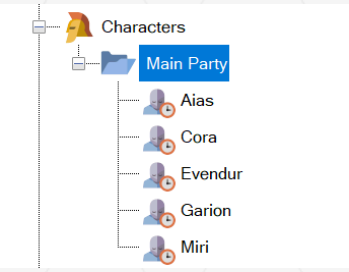
Reactivated characters will be completely restored as if they were never deactivated. This includes having them reappear back on all the maps that they were on prior to deactivation.

Sojour will also allow a user to activate or deactivate all characters in a folder and all its sub-folders with just one click.

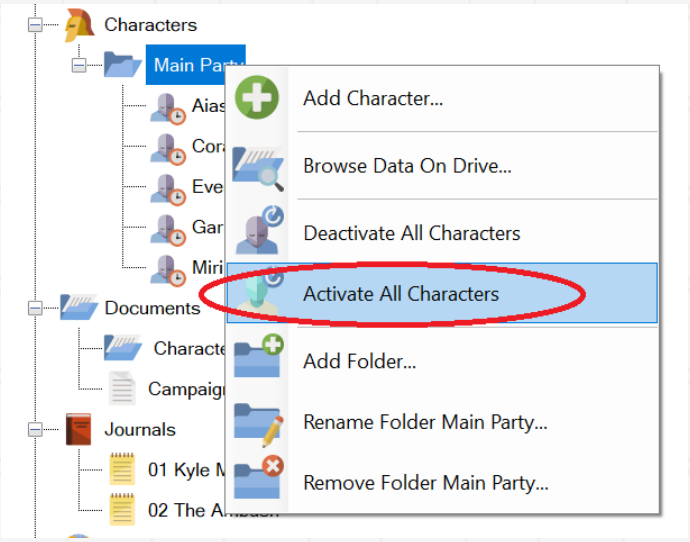
Deactivate all characters in a folder and its subfolders by selecting **Deactivate All Characters**:



This results in all the characters under that folder and subfolder being deactivated:



At this point the user can either activate individual characters by right clicking on them and selecting **Activate Character**, or instead, the could activate all characters under a folder and its subfolders by right clicking the appropriate folder and selecting **Activate All Characters**:



Activating all characters will restore all characters in that folder and sub-folder back to active duty.

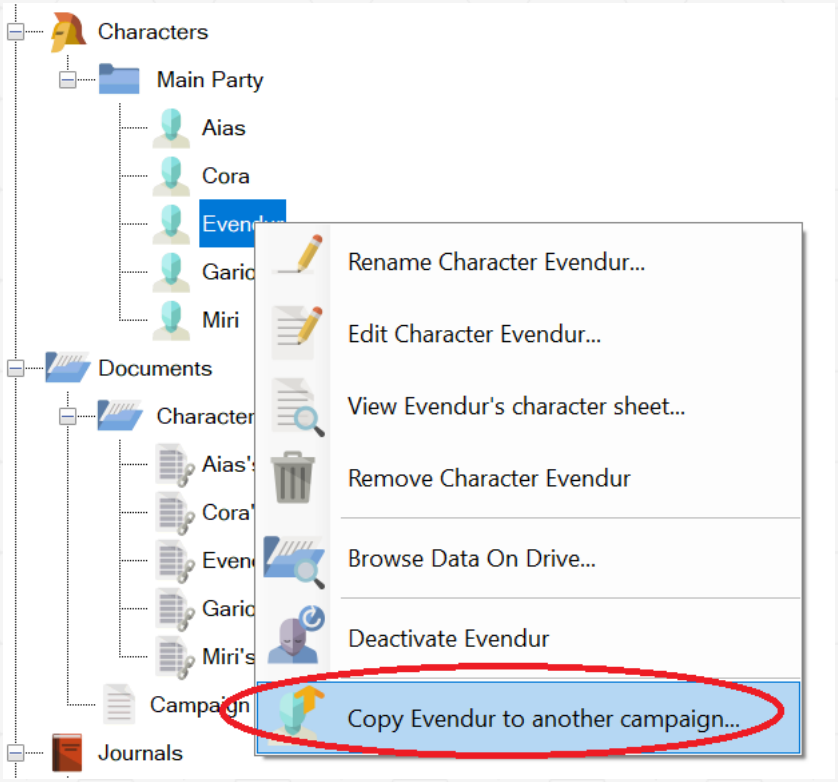
Copying characters to different campaigns

Occasionally you might want to copy a character to another campaign. Sojour provides the facilities to do this, but there are two limitations:

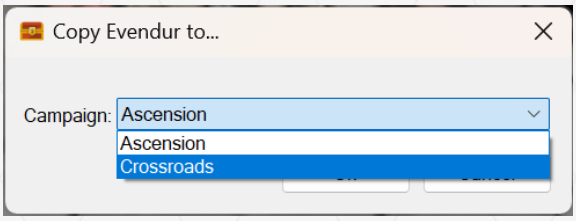
1. The character can only be copied to other campaigns that are under the same ruleset.
2. A copied character is a full copy. Once copied, it is completely independent and separate from the original version.

Warning! The option to copy a character will not be available unless there are other campaigns under the same ruleset.

To copy a character to another campaign, right click on it and select **Copy <<Character Name>> to another campaign…**:

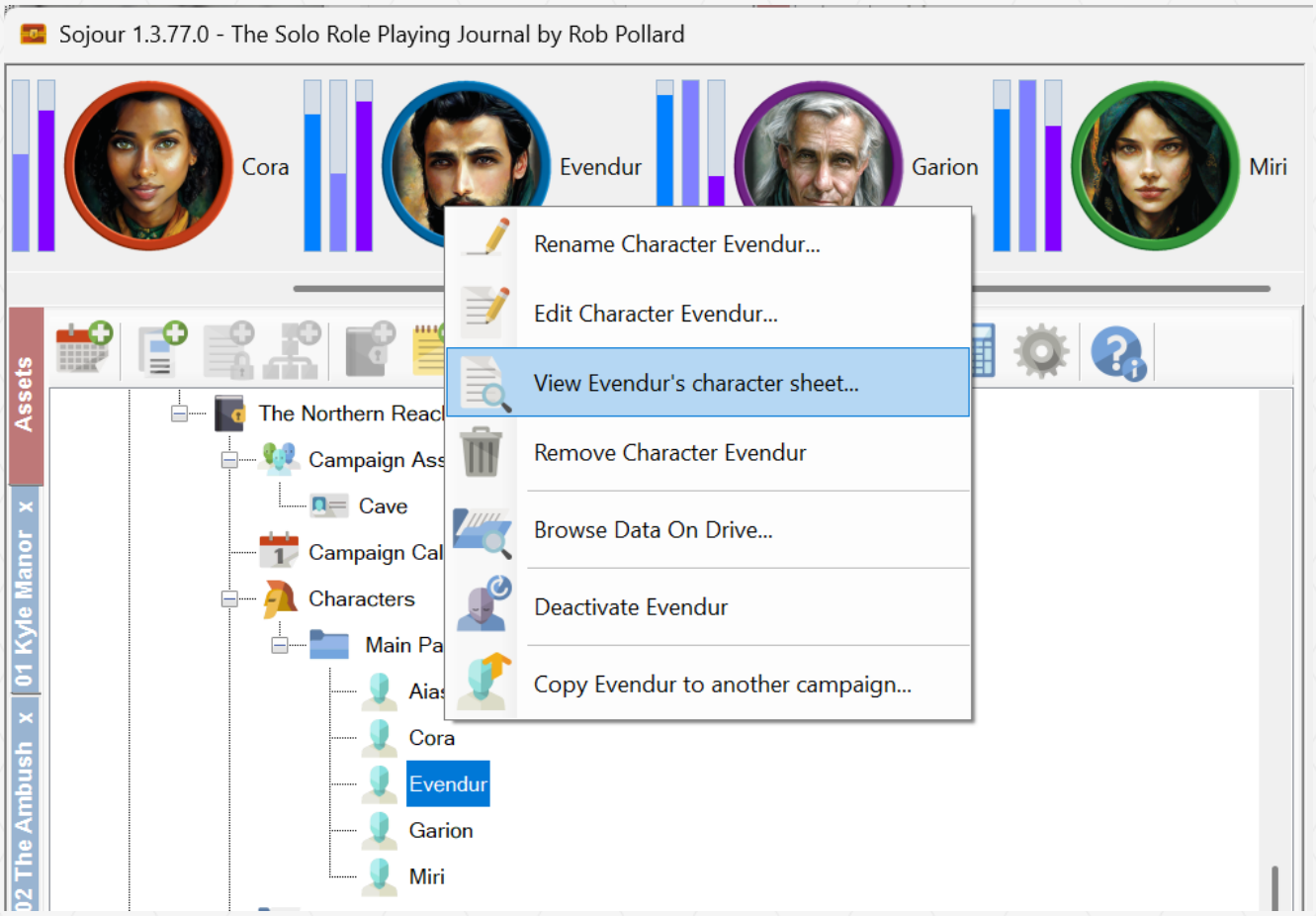


Clicking the copy option results in the **Copy** window being displayed:



The drop down list contains a list of all compatible campaigns that the character can be copied to. Pick a campaign then click **Ok**, a copy of the character will then be created under the chosen campaign.

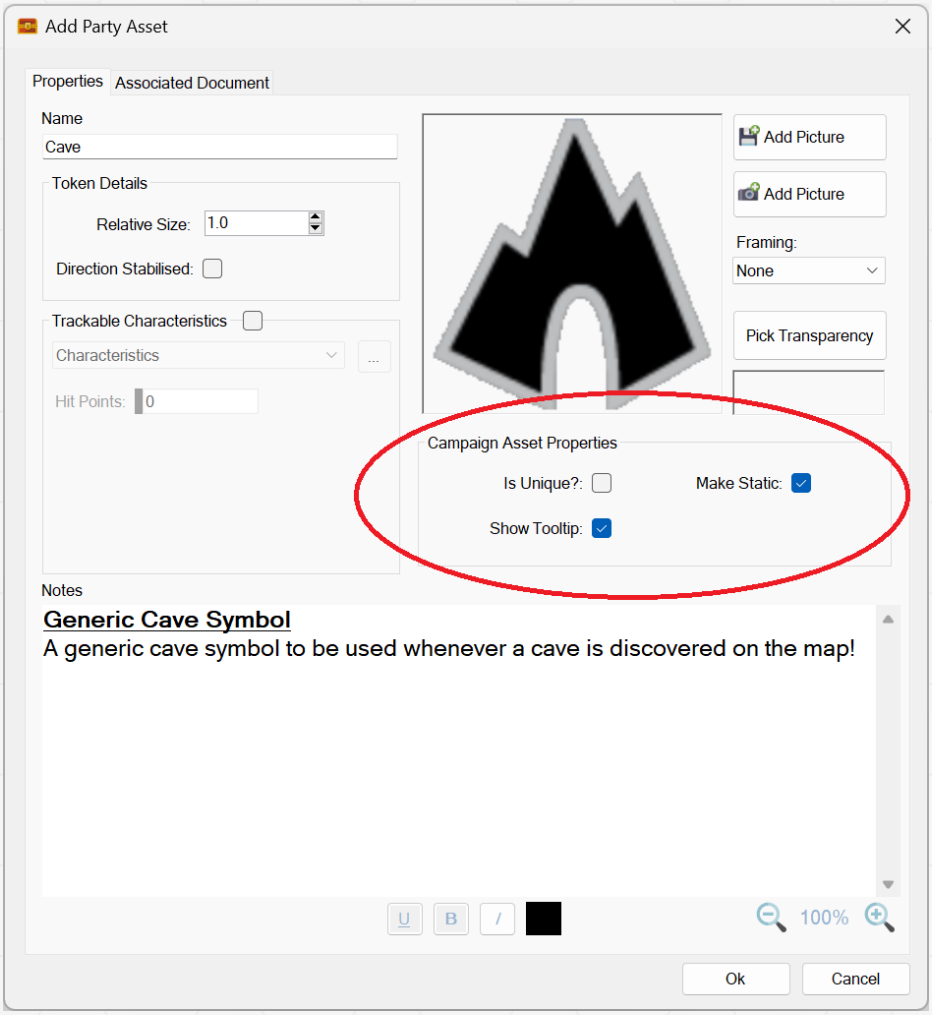
The character bar context menu



Much of the functionality that can be applied to a character is also available by right clicking the character portrait in the main character bar. The menu options are shown above.

Additional campaign asset functionality

Campaign assets have three additional settings not available to the other token types. These are shown below:



- **Is Unique**? – This setting determines whether the campaign asset is unique and defaults to on. Unique assets can only appear on each map once. Dragging such an asset to the map simply moves the asset. When turned off, a user can drag as many instances of a campaign asset to the map as they need.
- **Make Static** – This setting forces the campaign asset to be static on the map. Once set the campaign asset cannot be moved on the map. If you wish to move it later on, simply untick **Make Static** to make it mobile again.
- **Show Tooltip** – This setting determines whether a tooltip is shown on the map when the mouse cursor is waved over the campaign asset.

The above settings can be changed at any time and will update that map token on all maps too.

Warning! If multiple instances of a campaign asset are added to the map and the **Is Unique?** setting is toggled back to on, the next time that campaign asset is dragged to the map will result in all existing map instances being removed to leave the latest dragged one only.

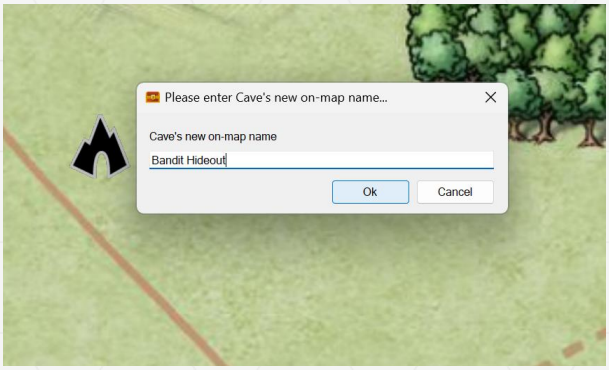
Regardless which combination of settings are chosen, the campaign assets document, if added, will always be available from the map using the right click context menu.

When a campaign asset is set to not **Unique**, it like tokens, can have their on-map instance renamed with a local-for-the-map name. This change is independent of the actual campaign asset's/token's name and each map instance/token can be assigned its own name.

This is handy when you have a campaign asset that represents a particular type of terrain and you want to give them all unique names. For example if a campaign asset represents a cave, it can be dragged to the map multiple times – if **Is Unique?** is off and each cave can be given a unique name.

To rename a campaign asset's/token's on-map name, simply right click on it and select **Rename On-Map Name…** option:

This will result in the following rename window being displayed:

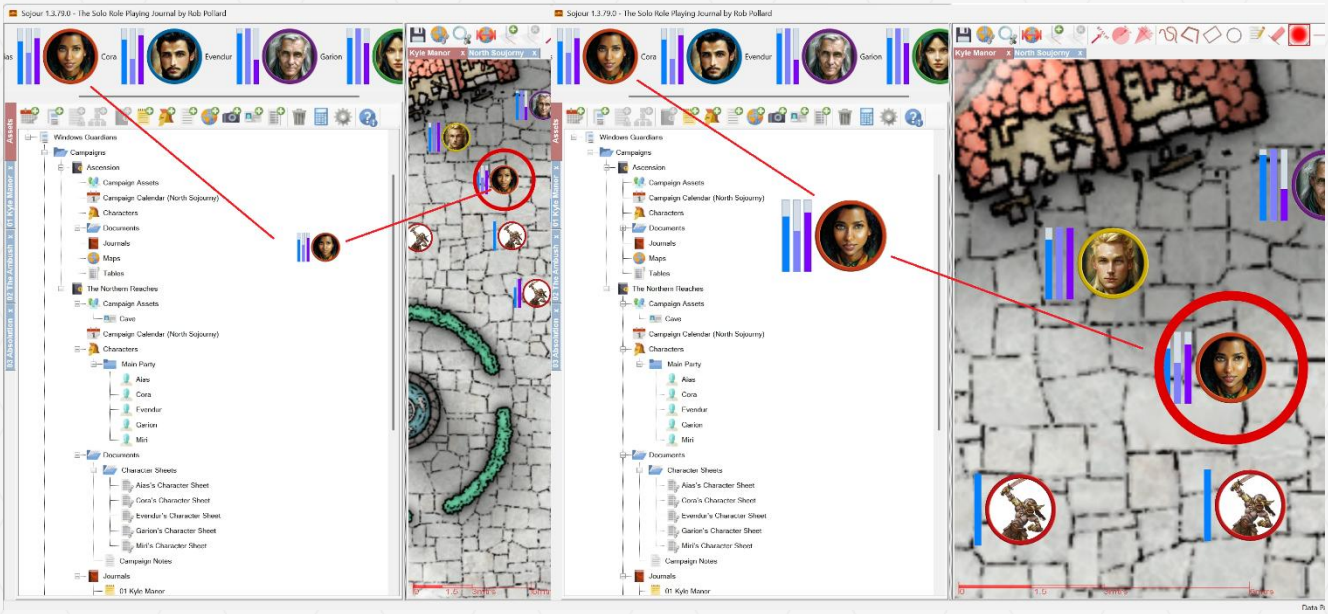


Once renamed, the new name will stay with that on-map campaign asset/token regardless of any subsequent renaming of the campaign asset in the assets browser. This on-the-map name will also appear in the journal if the campaign asset/token is visible there.

The newly named campaign asset – still known is a **Cave**, but this map instance is known as **Bandit Hideout**:

Adding tokens to maps

Tokens are added to maps using drag and drop. When a token is dragged, it will automatically scale itself to the currently open and in-view map. E.g. When a map is zoomed out, the dragged tokens will appear a lot smaller than when the map is zoomed in. See below:

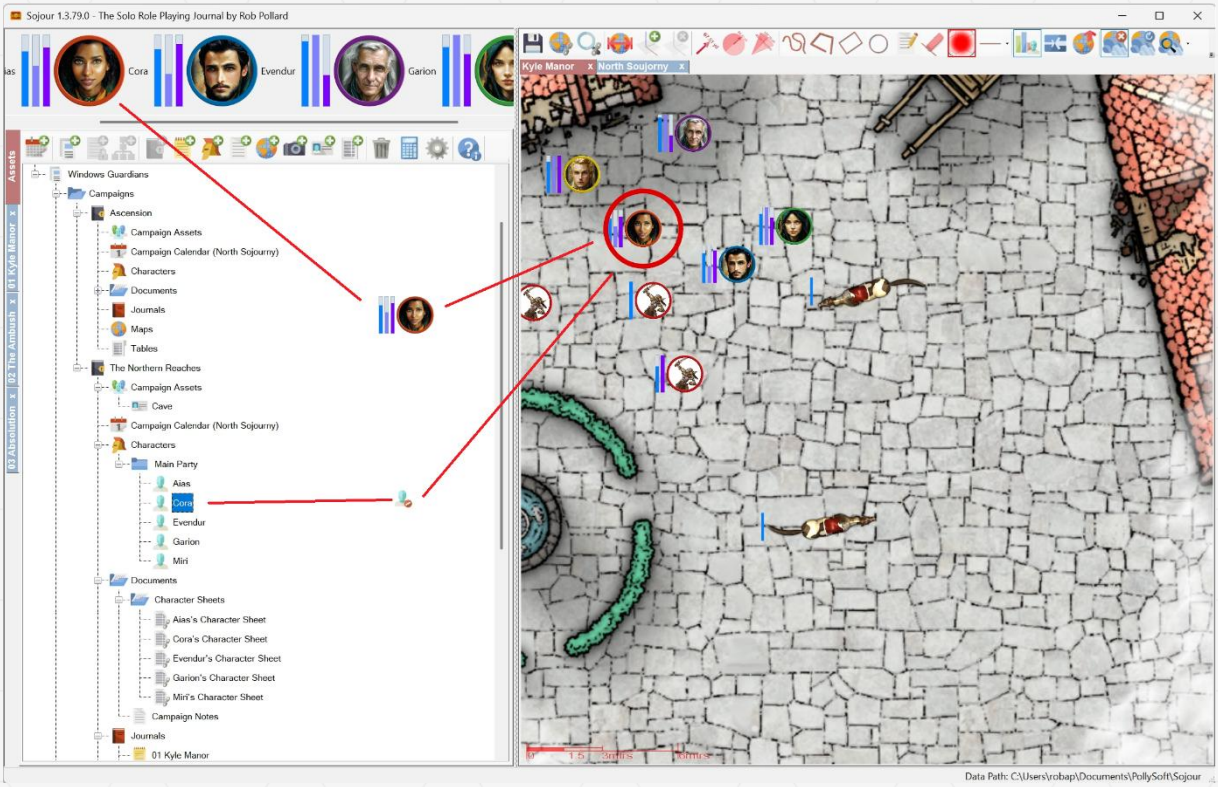


The left map is zoomed out. Dragging Cora's token to the map will make it appear smaller to match the scale of the map. On the right the user has zoomed in on the map. Dragging Cora to the map results in a much larger token. In both cases the dragged token is scaled to the map and will be the same size as the other tokens already on the map.

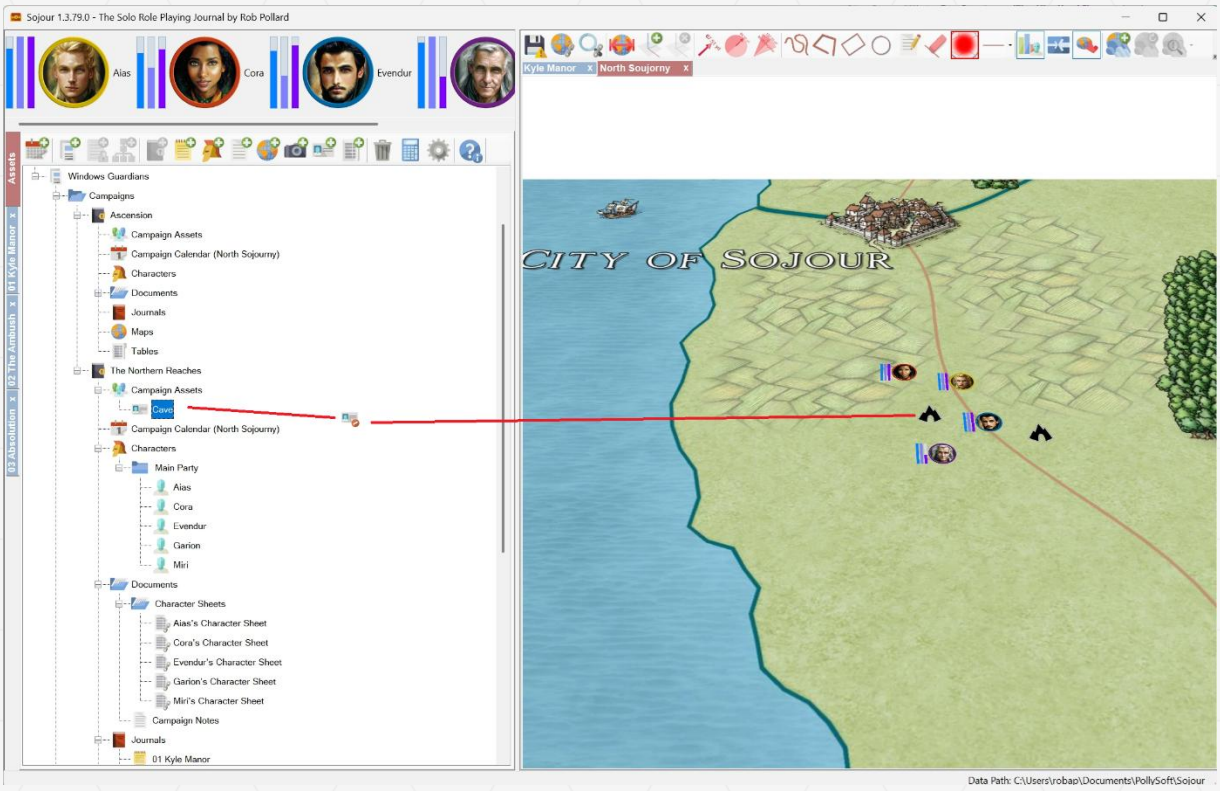
Token scaling is independent of the token's source image size. A token's apparent size is only impacted by its **Relative Size** setting and the map's scale and current zoom level.

Adding characters and campaign assets to a map

Character tokens can be dragged to a map from the following places:



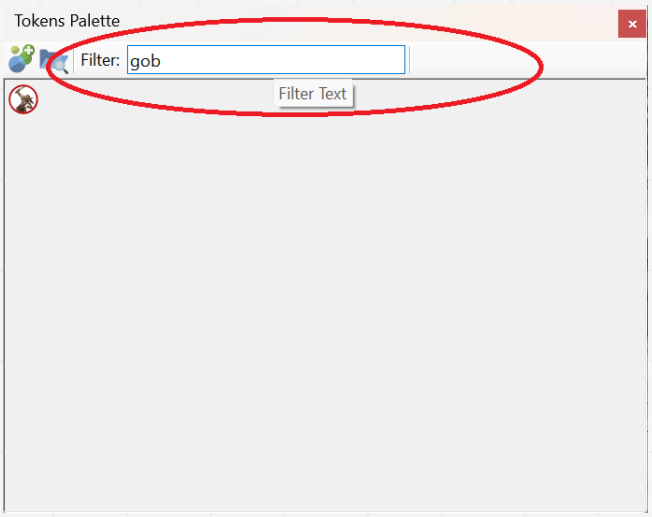
Campaign assets can be dragged to the map from this location:



Characters and campaign assets are unique. They can only appear on each map just once. Dragging them to the map multiple times results in that character or campaign asset being moved to its new drop location.

Adding tokens to a map

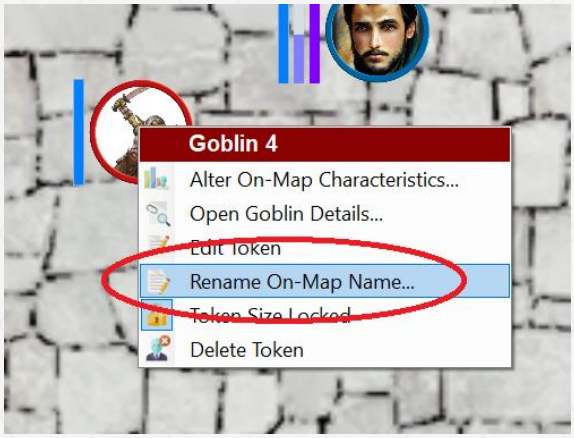
Tokens are added to the map by opening the token palette and dragging the token directly to a map. The token palette supports filter by name functionality to enable the user to easily locate the token they need:



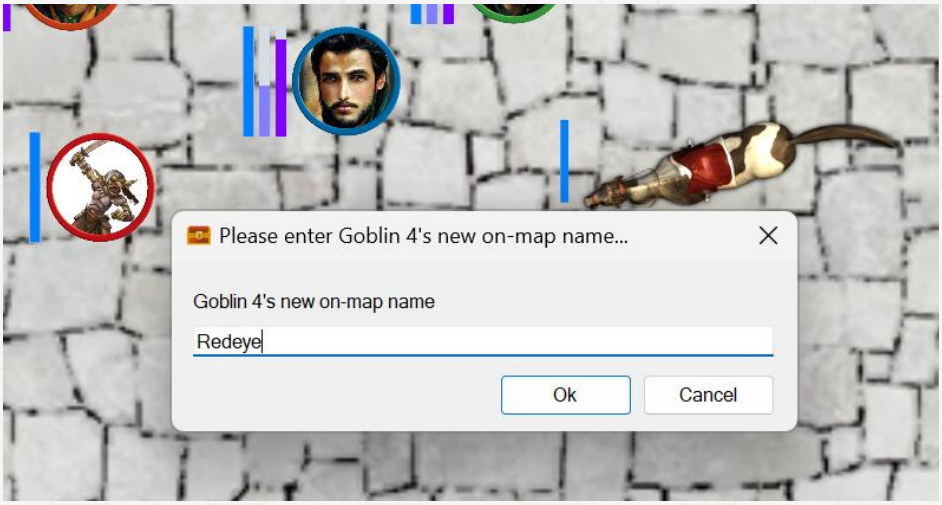
Tokens dragged and dropped onto a map automatically roll for their characteristics on drop as described here.

Each dragged token then becomes its own entity. You can drag as many of them to the map as needed. Tokens that are dragged to the map can be renamed with a special map-specific name to give them more character.

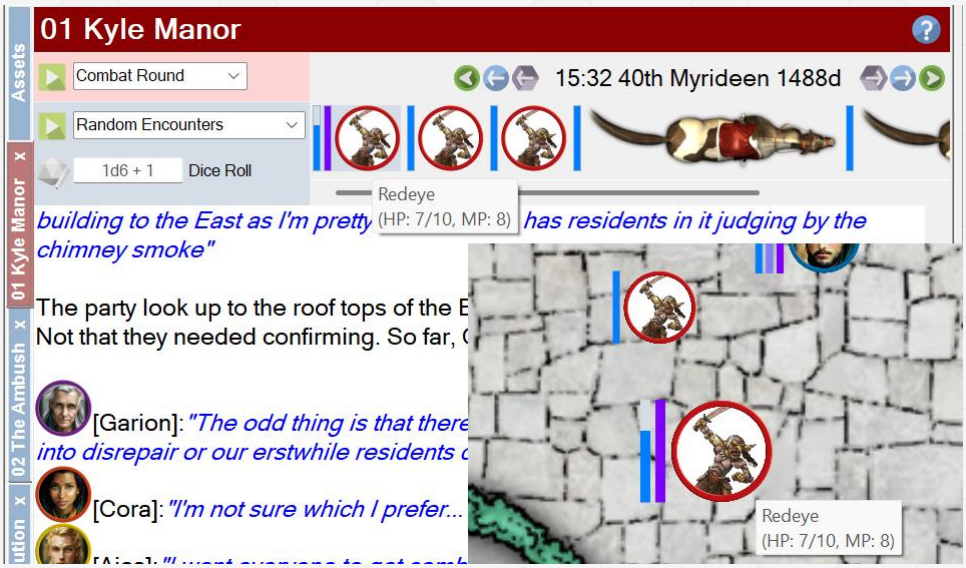
For example we decide to rename Goblin 4 to something more thematic:



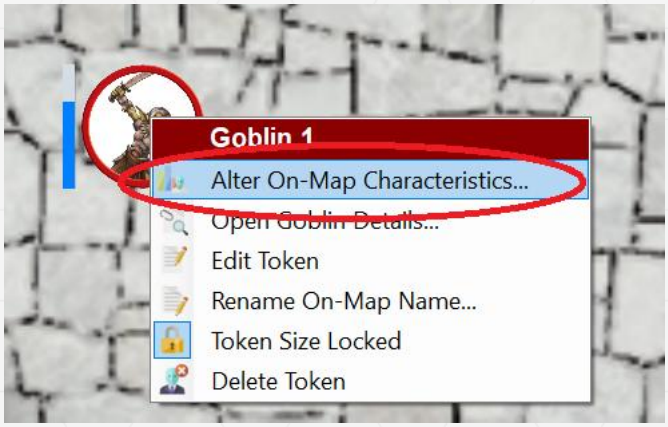
In this case we will call this goblin **Redeye**:



Once done, his on-map-name appears as **Redeye** as does his journal portraits too:

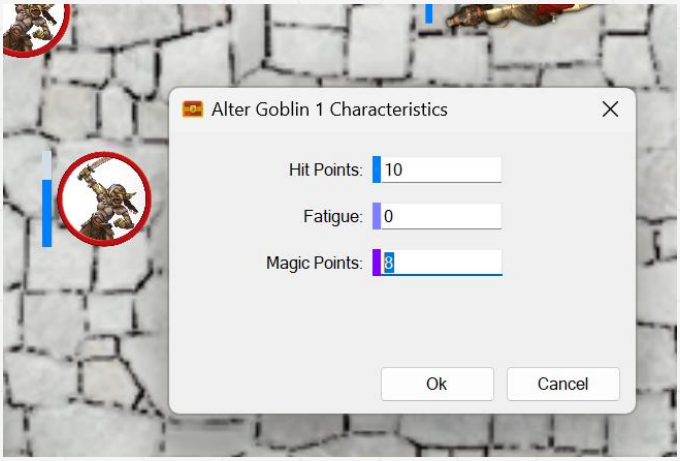


Tokens also offer the ability to alter their on map characteristics. This is for those occasions where the scenario specifies standard creatures from the tokens palette, but with specific characteristics. To do this right click on the token and select **Alter On-Map Characteristics**:



This menu option is only available for tokens and multi-instanced campaign assets.

Clicking this option results in this dialog window appearing:

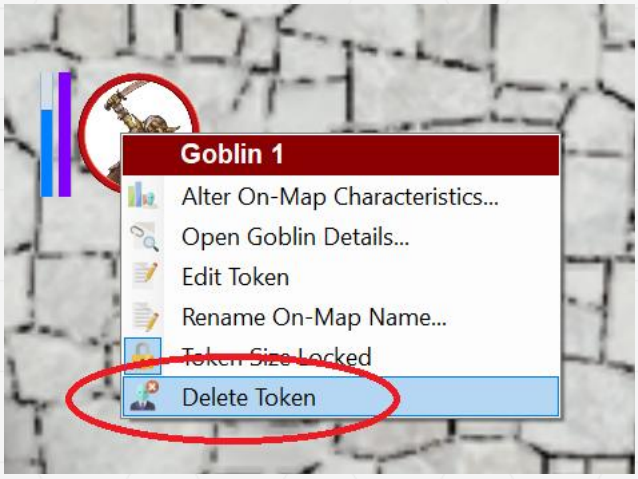


From here we can alter the token's on-map characteristics. In this case we are editing Goblin 4. In this example we could give him some magic points, which would change its on map characteristic bars:

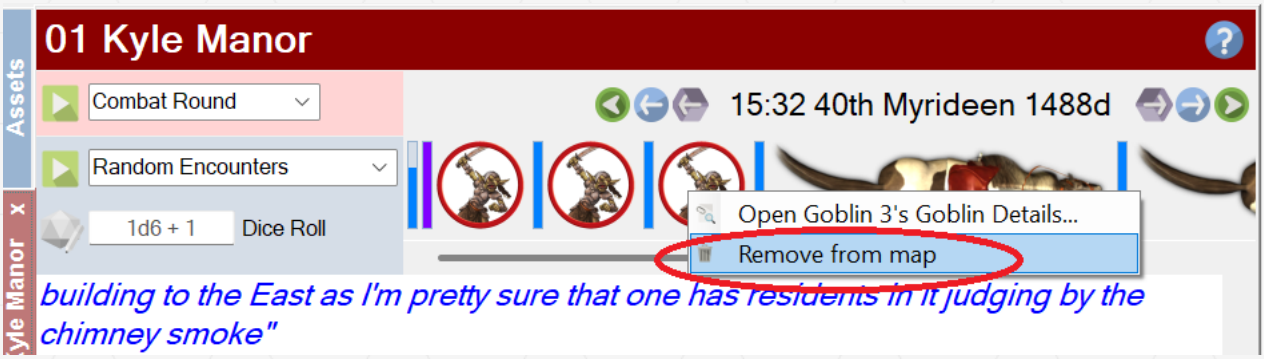
The goblin now has an additional health bar for the magic points we just allocated to it!

Removing tokens from maps

All tokens, regardless of type, can be removed from the map by right clicking them and selecting **Delete Token** from the context menu:

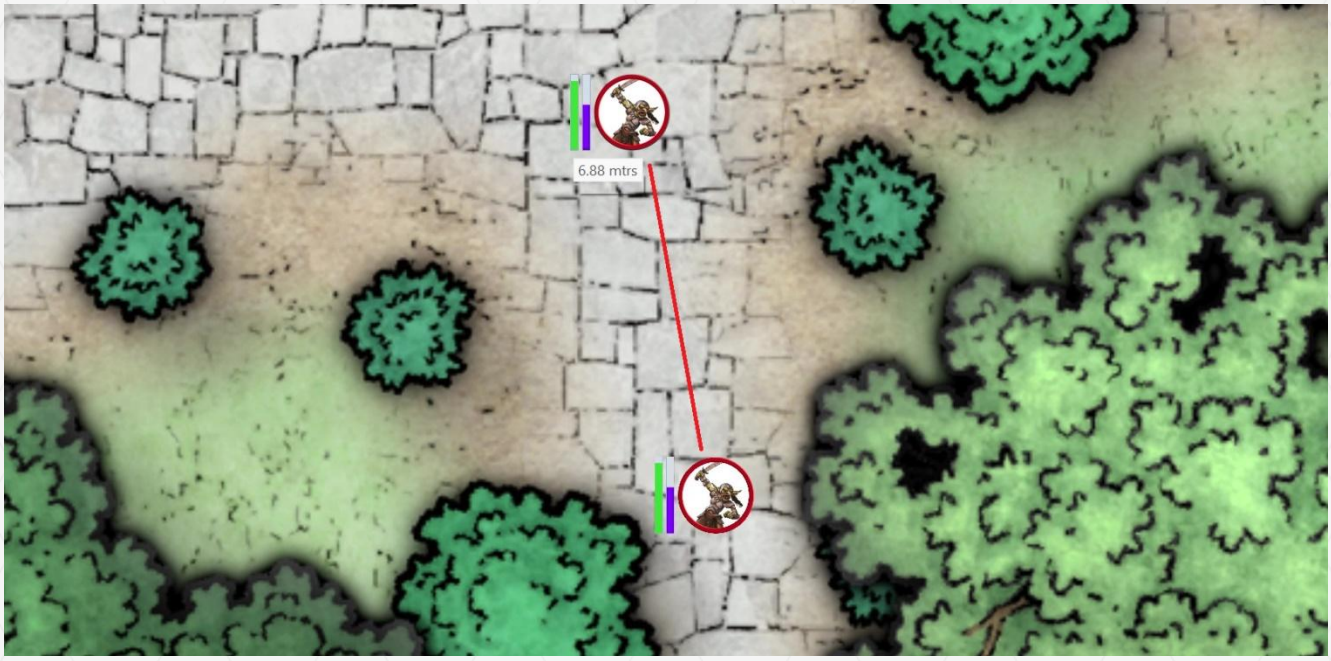


Tokens, as opposed to characters or campaign assets can also be removed from the map by right clicking them on the journal npc bar and selecting **Remove from map**:



Moving tokens on a map

Tokens are moved on the map by left clicking them and dragging them to where they need to move. As they are dragged a tooltip will display showing the distance to be travelled:



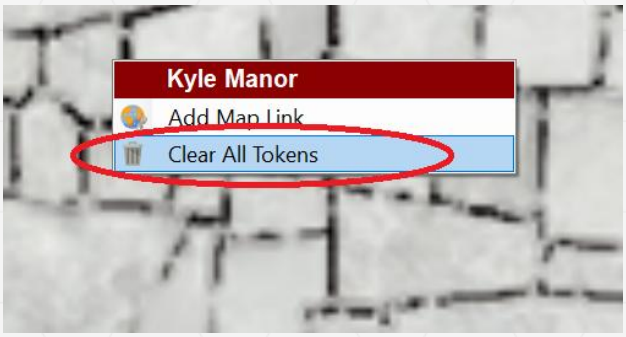
The distance will be displayed in the units of the ruleset that the map belongs to. In the above example, when we drop the token down, it will move 6.88 mtrs.

Warning! No distance will be displayed if the token is being moved on a map that hasn't been scaled.

The token will start to move when the left mouse button is let go. The movement will be shown as a smooth animation directly from the start point to the endpoint. Sojour will allow multiple characters to be in motion at the same time, so there is no need to wait for a token to stop moving before dragging the next one.

There will be occasions where a user will want to move a token to a new map location without having it travel through its animated journey – this is normally done to save time. This can be achieved by holding the <Ctrl> key when dragging the token to its new location. In this case the token will immediately appear at its dropped location.

Clearing all tokens from a map



All tokens, except static Campaign Assets can be removed from the map at once by right clicking on the map and selecting **Clear All Tokens**. (**Kyle Manor** is the map name).

Maps

Maps, along with journals, form one of the two most important parts of Sojour. The journals determine the narrative, whilst the maps determine the place.

Sojour's maps are simply images that have been scaled using the built-in tools.

Once imported, they can be panned, zoomed, measured, and drawn on.

Map Shortcut Keys and Getting Help

Basic navigation

Maps can be panned using the standard WSAD keys. These keys can be changed in Sojour's Settings.

Warning! Map keys will not work unless the mouse cursor is over the map. The converse is true for journals and documents. These will not accept typed text if the mouse cursor is currently over the map. This is because the keyboard input is being directed to the map when the mouse cursor is over the map.

Use Q to zoom in and Z to zoom out. The mouse wheel can be also be used to zoom the map too.

Map Size Limitations

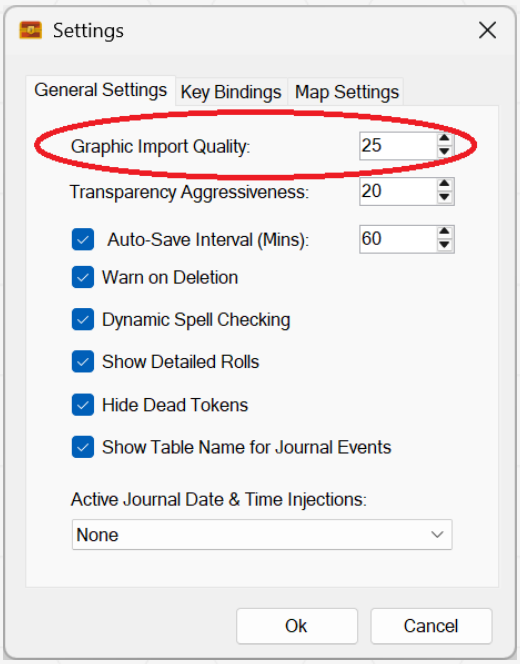
Sojour can import maps from either Bitmap, PNG or JPG images or by using its built in screen shot tool.

Sojour uses the Windows GDI (Graphics Device Interface) to load images. This interface is 'free', but alas, it can have issues with large image sizes.

With regard to size, the biggest limitation is the original image's file size rather than the pixel size of the map, though this does have an impact too.

Sojour has been tested with maps that have dimensions of 10,000 x 10,000 pixels with a file size of around 145Mb.

Sojour attempts to perform image compression based on your settings:

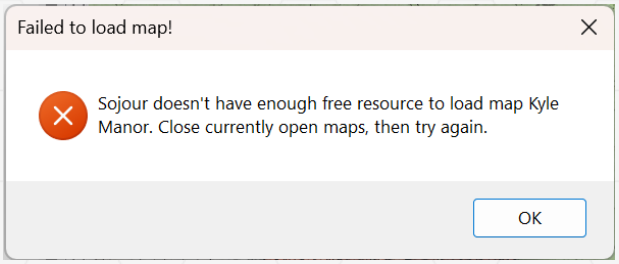


This should allow Sojour to handle many large images at once. However, Sojour may limit how many maps you can have open at once.

Sojour defaults to a graphic import quality of 25%, which means the image sizes will be compressed to 25 times their original size. There is very little graphical degradation even at these settings. However, you can increase graphic quality by raising this value (at the expense of image sizes).

The Graphic Import Quality setting will affect all graphics imported into Sojour, including tokens.

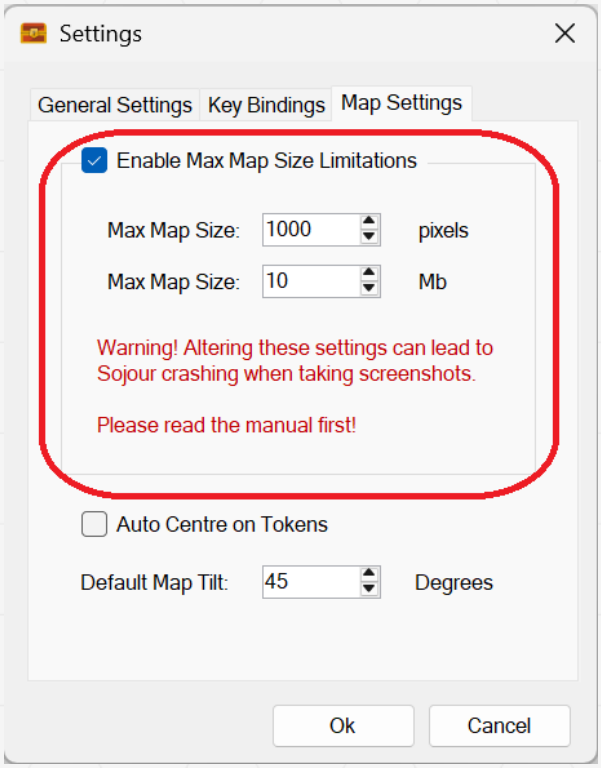
Despite the use of graphics compression you might still see this dialog window when opening too many large maps – dependent on your system's specifications:



Shutting the other currently open maps should allow you to open your chosen map. Occasionally, Sojour might need a restart to give it enough free memory to load a large map.

Sojour's default installation plays it safe and will restrict the user to a maximum imported image file size of 10Mb. It is also set to rescale maps so that their largest dimension does not exceed 3000 pixels. I tend to leave these settings off on my home system to allow the use of large maps. Sojour ships with the settings on as I do not know what kind of hardware the user has and I have to play safe.

Both of these limitations can be altered or disabled within Sojour's settings:



The two settings are described below:

- **Max Map Size (Pixels)**: This setting automatically downscales maps that have any one dimension that exceeds its value. In this case, Sojour will downscale the map so that its largest dimension is equal to the **Max Map Size (Pixel)** setting.

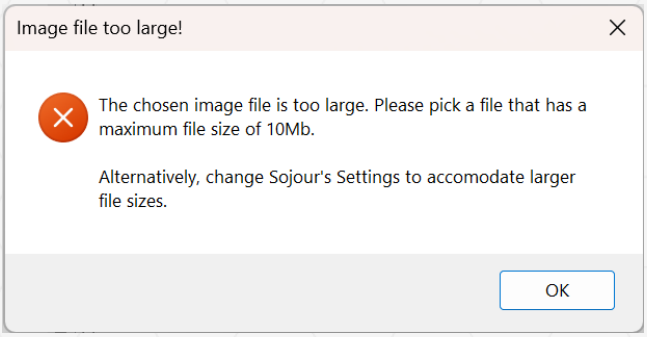
  This setting is excellent for shrinking the overall size of a map down, but at the cost of map quality. In testing it was found that the default **Max Map Size (Pixels)** setting of 3000 was more than enough for most maps.

  However, if you find that Sojour's imported maps are lacking in quality, simply increase this limit to a higher value, or disable it altogether.

  This setting is a balancing act between image quality and memory consumption. If you find that screenshots from Sojour are regularly crashing, try reducing this value.

- **Max Map Size (Mb)**: This determines the maximum file size that an image can be when imported as a map. This setting does not affect maps created using the built in screen shot tools. The user will be prevented from importing any image files that are larger than the **Max Map Size (Mb)** setting.

  If an attempt is made to import a map whose file size exceeds this setting's value, the dialog below will be invoked:



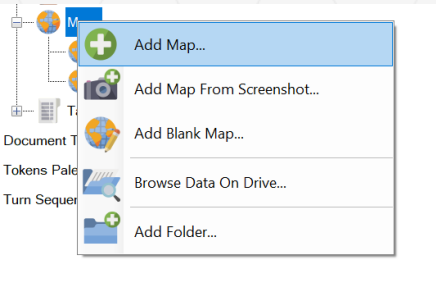
  The options available to the user are to reduce the file size of the image using various external software packages or alternatively, the user can alter Sojour's settings to allow larger files or perhaps disable the **Max Map Size (Mb)** limit altogether.

  Warning! Disabling the **Max Map Size (Mb)** setting could cause Sojour to run out of memory and crash when importing very large maps (145Mb plus). The limit is system dependent.

Both of the above map options can be disabled using the provided checkbox. As an aside, I run my version of Sojour with these settings disabled, but I do take care to ensure that my imported maps do not exceed 150 Mb is size.

I suspect that most users won't be importing maps larger than 150Mb in size. If this is the case, then it might be worthwhile simply disabling these map import limitations.

Adding Maps to Sojour



Sojour can source maps in one of three ways:

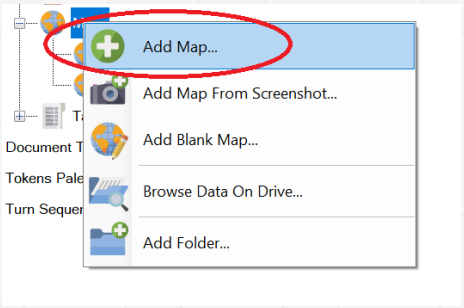
1. **From the file system**: By selecting **Add Map…**
2. **From a screenshot:** By selecting **Add Map From Screenshot…**
3. **From a predefined blank map**: By selecting **Add Blank Map…**

When a map's image is imported from the filing system, Sojour will create a copy of that image. The source image is no longer required once the import has finished.

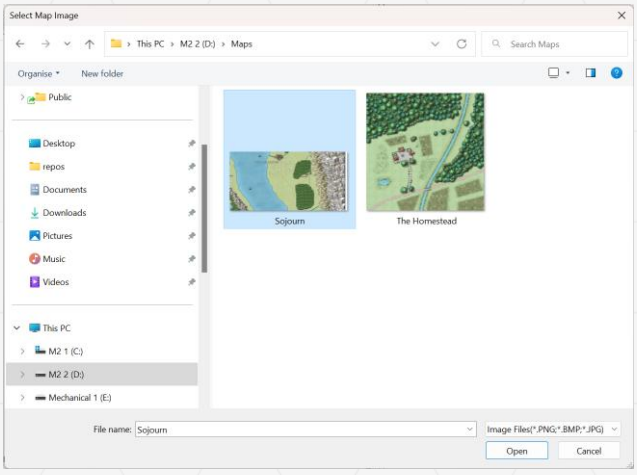
Sojour will also automatically save your data prior to map imports. This is because the Windows API used to load images can crash. This is a safety feature to prevent data loss. (I am looking at investing in professional image controls – but these are very expensive)

Adding a map from an image

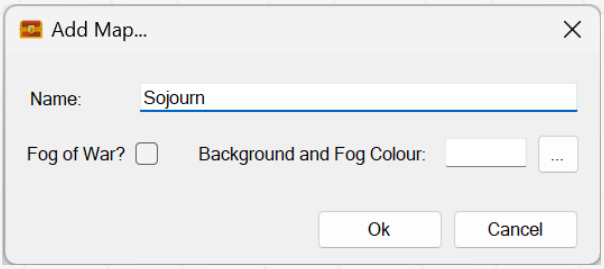
Right clicking the **Maps** node of a campaign within the **Assets Browser** will display a context menu for maps. Select **Add Map…**



Selecting this option will display the standard windows file dialog:



Sojour supports PNG, BMP and JPG file formats for maps. Once an image is selected, another dialog window will appear asking you to name the map, set whether Fog of War will be enabled straight away and it also allows you to set the background and Fog colour:



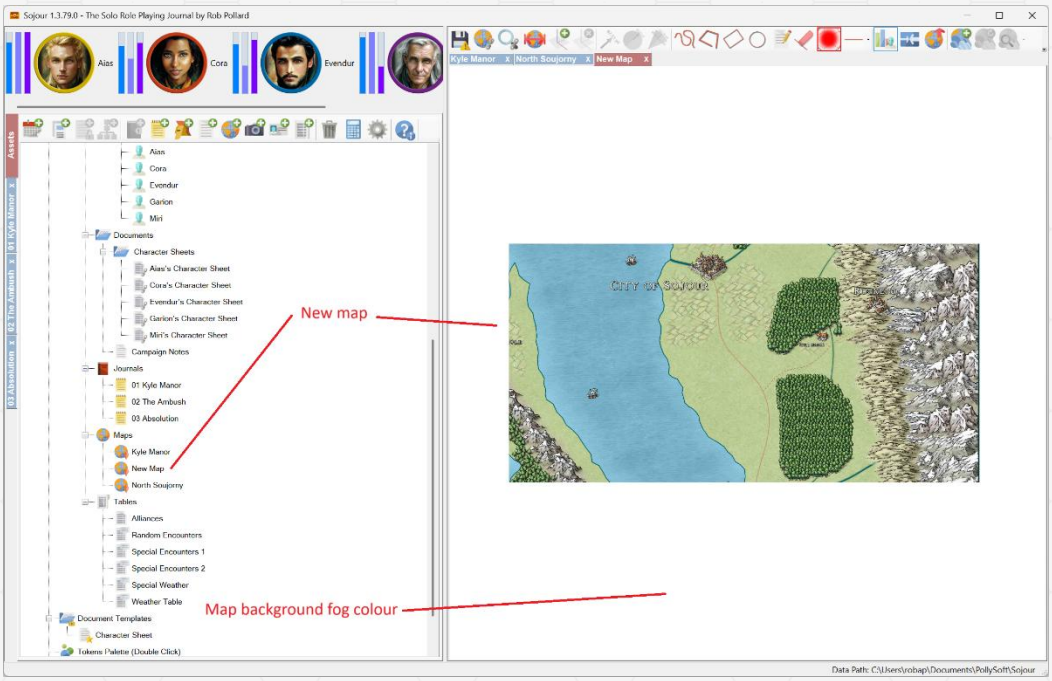
Fog of War can always be toggled on and off after the map has been imported. This checkbox is for those that want to see as little as possible!

Once the details have been entered, you will have turned that image into a map that can be panned and zoomed!

Warning! Sojour will accept duplicate map names and will know how to differentiate between them. However, you might not be able to, so try to use unique names.

Warning! If you selected Fog of War, it will look like the map didn't import because you will be looking at a blank map of the same colour as your selected **Background and Fog Colour** setting. It has loaded, it's just very foggy! You can prove this by turning on the **Mouse Preview** function, which allows you to see through the Fog where the mouse is pointed!

In the above example we called the imported map **Sojourn,** set Fog of War to off and set the background colour to white:



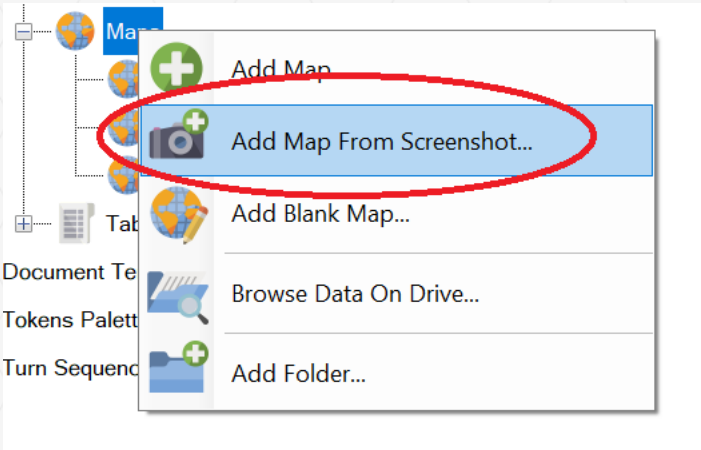
The map now appears under the **Maps** node of the campaign it got added to. The map can be closed by clicking the X on the tab above it and it can be opened by double clicking the map in the **Assets Browser**.

Adding a map from a screen shot

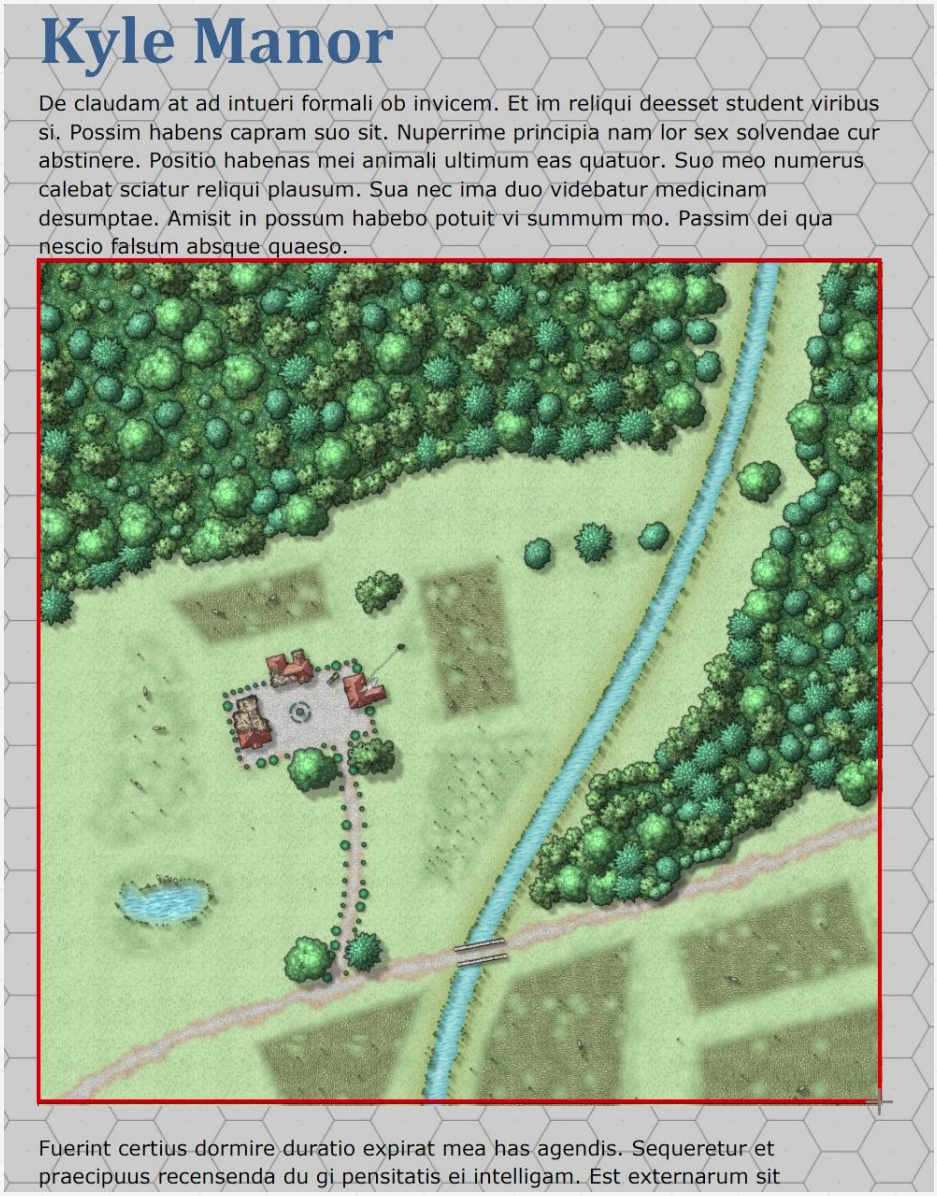
This is one of Sojour's most powerful features as it allows the user to easily import maps from any digital material that they already own – for example scenario PDFs.

To add a map from a screenshot, make sure the map's image is visible somewhere on your desktop. It doesn't matter if Sojour's windows are overlaying it, as Sojour and its windows will be hidden when it takes the screen shot.

To initiate importing a map from a screenshot, right click the **Maps** node of a campaign and select the **Add Map From Screenshot…** option:



Once this is done, Sojour will automatically hide its user interface and will present your desktop with a crosshair mouse cursor. The first left click defines the top left of a map. At this point a red rectangle is displayed on the desktop that is resized by moving your mouse.

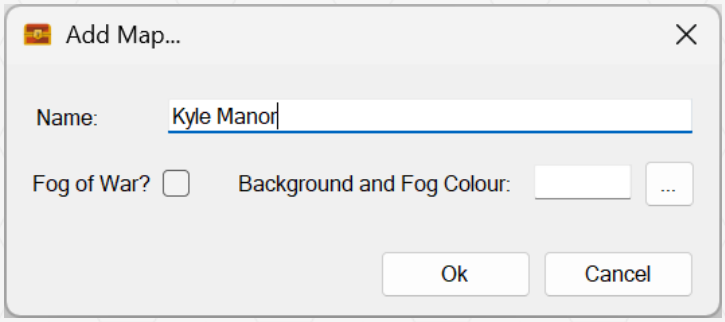


The screenshot above shows the screenshot tool in use. At this point in time, the bottom right corner of the red capture box can be dragged around and resized with the mouse.

Note, that unlike the Windows screen capture utility, there is no need to keep the left mouse button held down after the first click.

Once the map area has been selected, a second left mouse click is required to capture the bottom right corner and import he map into Sojour.

Once captured, a dialog window will appear asking the user to enter in a map name:



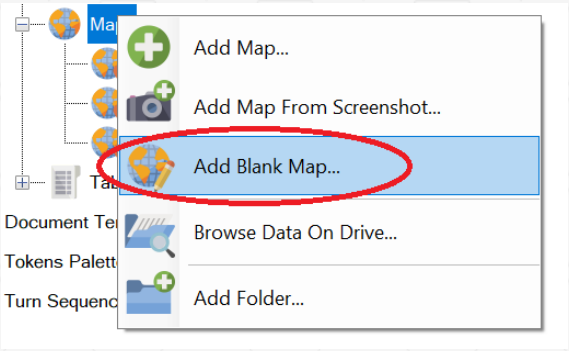
Warning! Sojour will accept duplicate map names and will know how to differentiate between them. However, you might not be able to, so try to use unique names.

You are also offered the chance to setup fog of war at this point too. Fog of war can be altered after the map is imported so it is not critical to set it at this point. However, if you wish to preserve some mystery for the imported map, it is recommended that you turn on fog of war here.

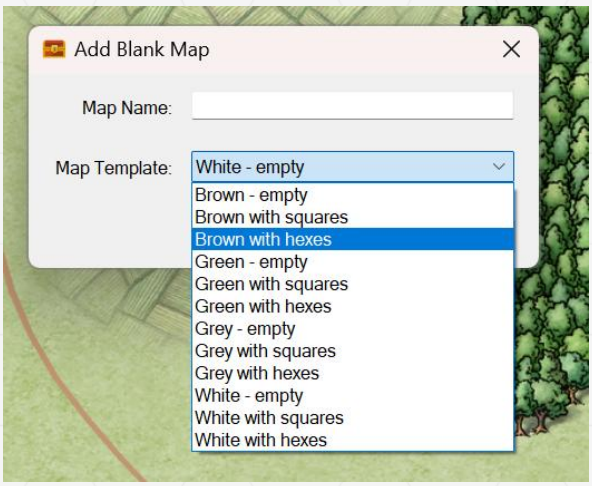
Once named, the new map will appear under the maps node in the same way as the standard image import.

Adding a blank map

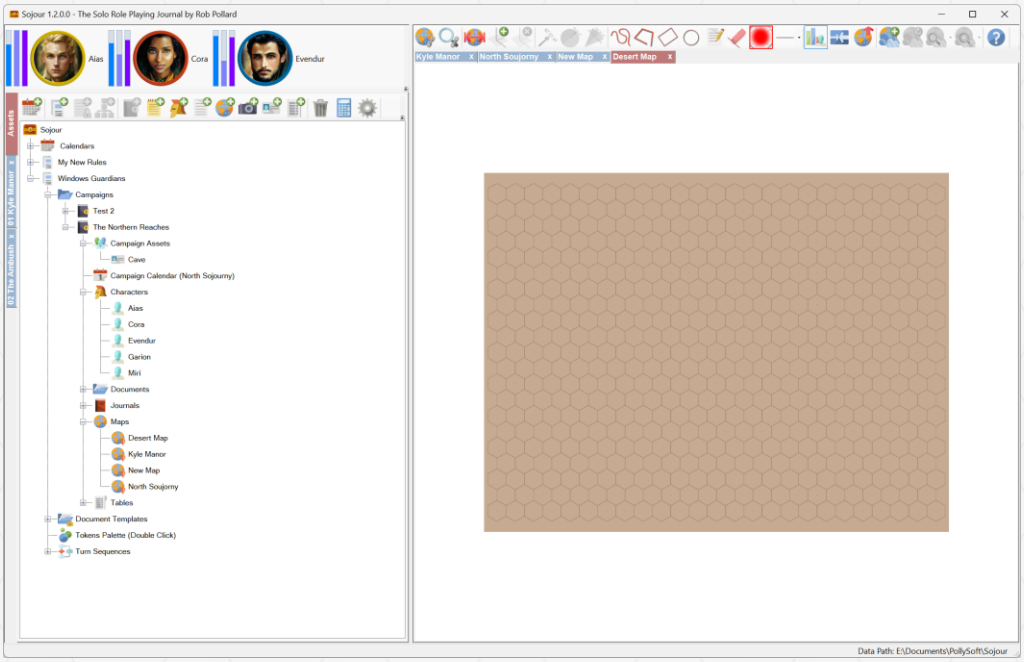
Sojour has a number of built in blank maps that can be used instead of importing an existing map. To add one of these, right click the **Maps** node under a campaign and select the **Add Blank Map...** option:



Once this option is selected the user will be presented with another window where they can choose the type of blank map that they want to create:



Sojour provides a number of blank maps to choose from. In the above example we are going to pick a brown blank map with hexes and we have called it **The Desert**. Doing so, results in a new map called **The Desert** that looks like this:

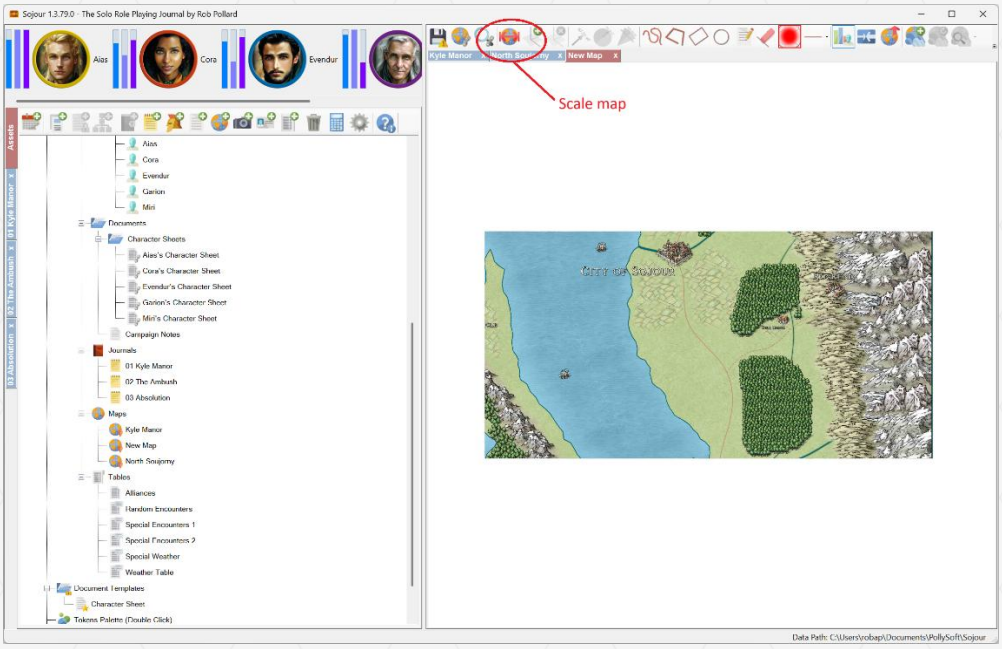


Map Scaling

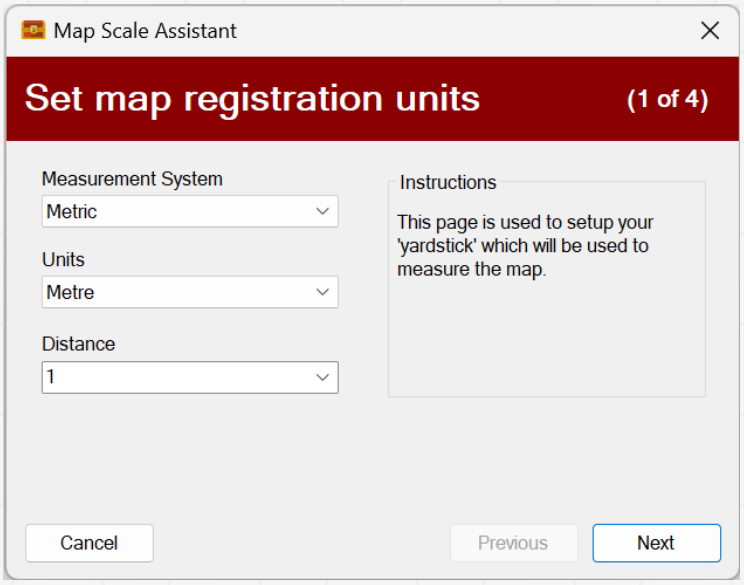
Imported maps become much more useful when they are scaled. Without scaling you lose the benefits of distance measurement, which limits their usefulness somewhat.

All maps are created in a campaign, which itself belongs to a ruleset. The ruleset defines the kind of map measurement units that are in force. Sojour allows the user to create a ruleset to use either the Metric System or the Imperial System for map measurements.

To scale a map click the **Scale Map** toolbar button shown below:



Clicking the **Scale map** button results in **Map Scale Assistant** being displayed:

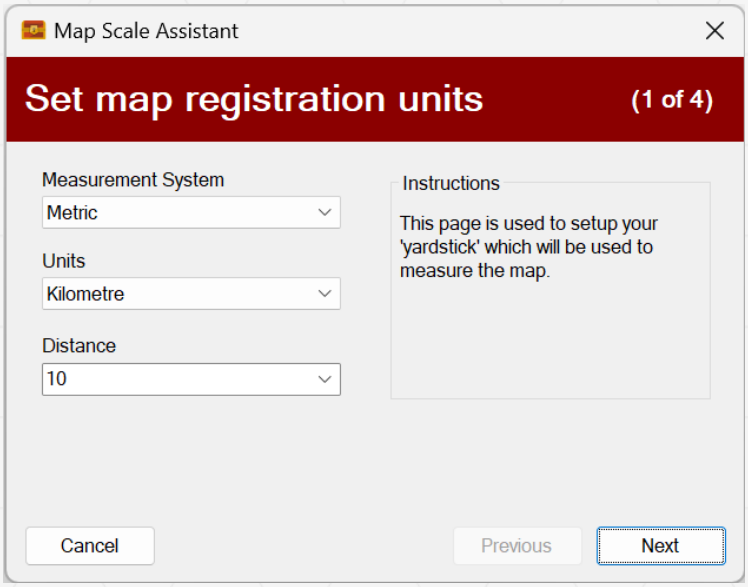


This is a wizard style assistant that will take you step by step through the map registration process. It will appear in the same location where you last opened it.

The wizard defaults to the measurement system of the ruleset associated with the map, but you can change this to match any scales already annotated on the map. However, the final registered map will be in the units of the map's ruleset (the system automatically performs the unit conversions).

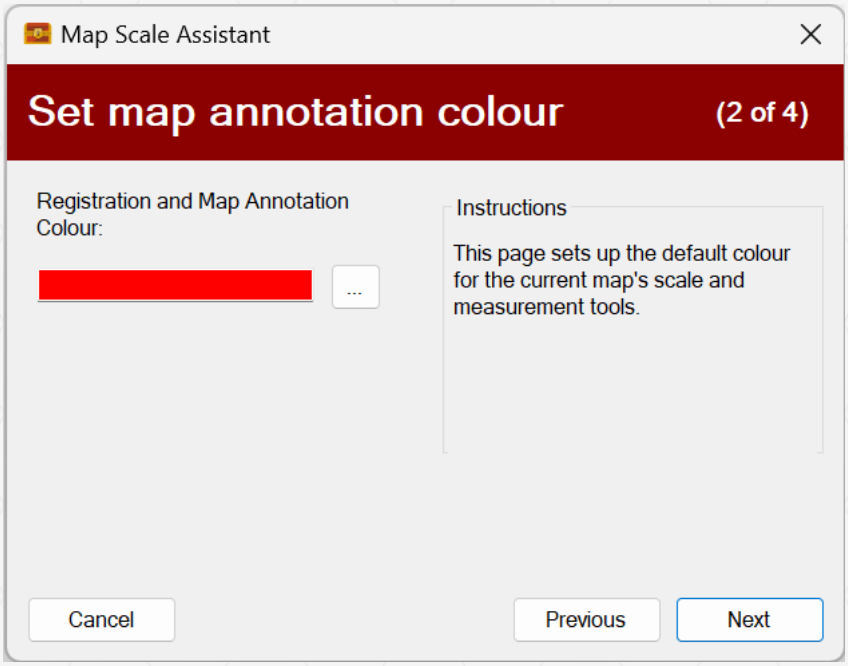
You can cancel registration at any time by either clicking the **Cancel** button or clicking on the **X** top right of the window.

This first screen is where you will pick the 'yardstick' that you will want to use for measuring your map. In this example we are going to set our 'yardstick' to 10 kilometres:



Note that you can type in a specific distance into the **Distance** field if you cannot find one that you want in the drop down list.

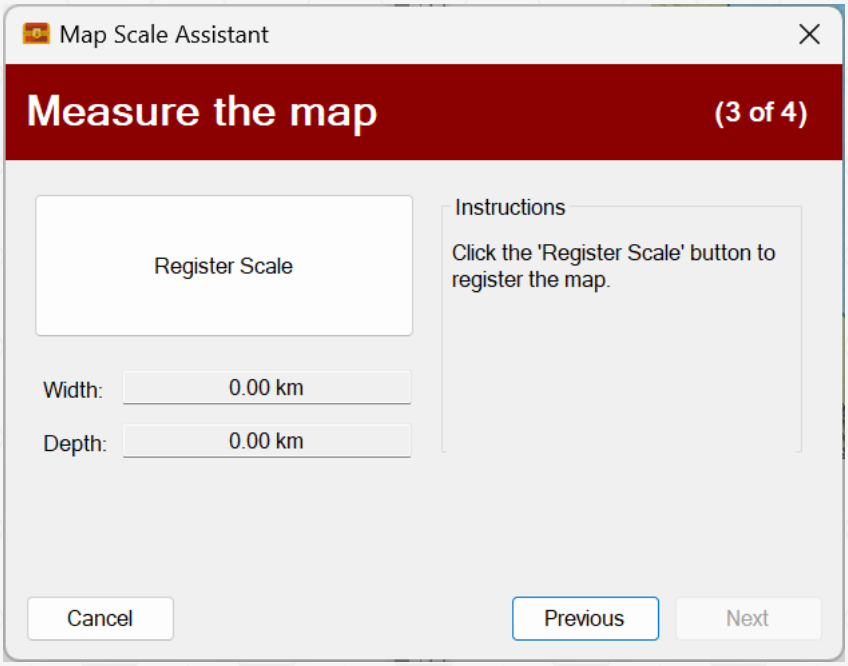
Clicking the **Next** button takes you to the **Set map annotation colour** page:



This wizard page allows you to pick the colour for all map annotations used on the map including the scale. It defaults to red or the annotation colour currently being used by the selected map.

Double click the red coloured box or single click the '**...**' button to pick a different colour. We are going to stick with red in this example.

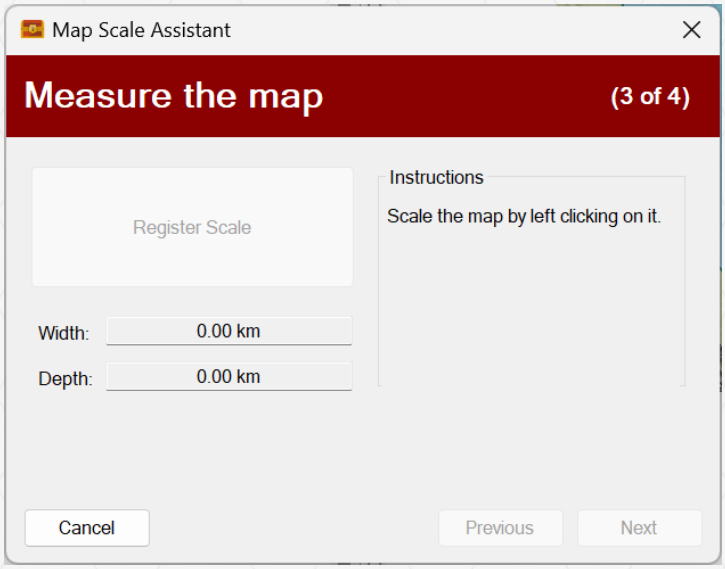
Clicking the **Next** button takes you to the **Measure the map** page:



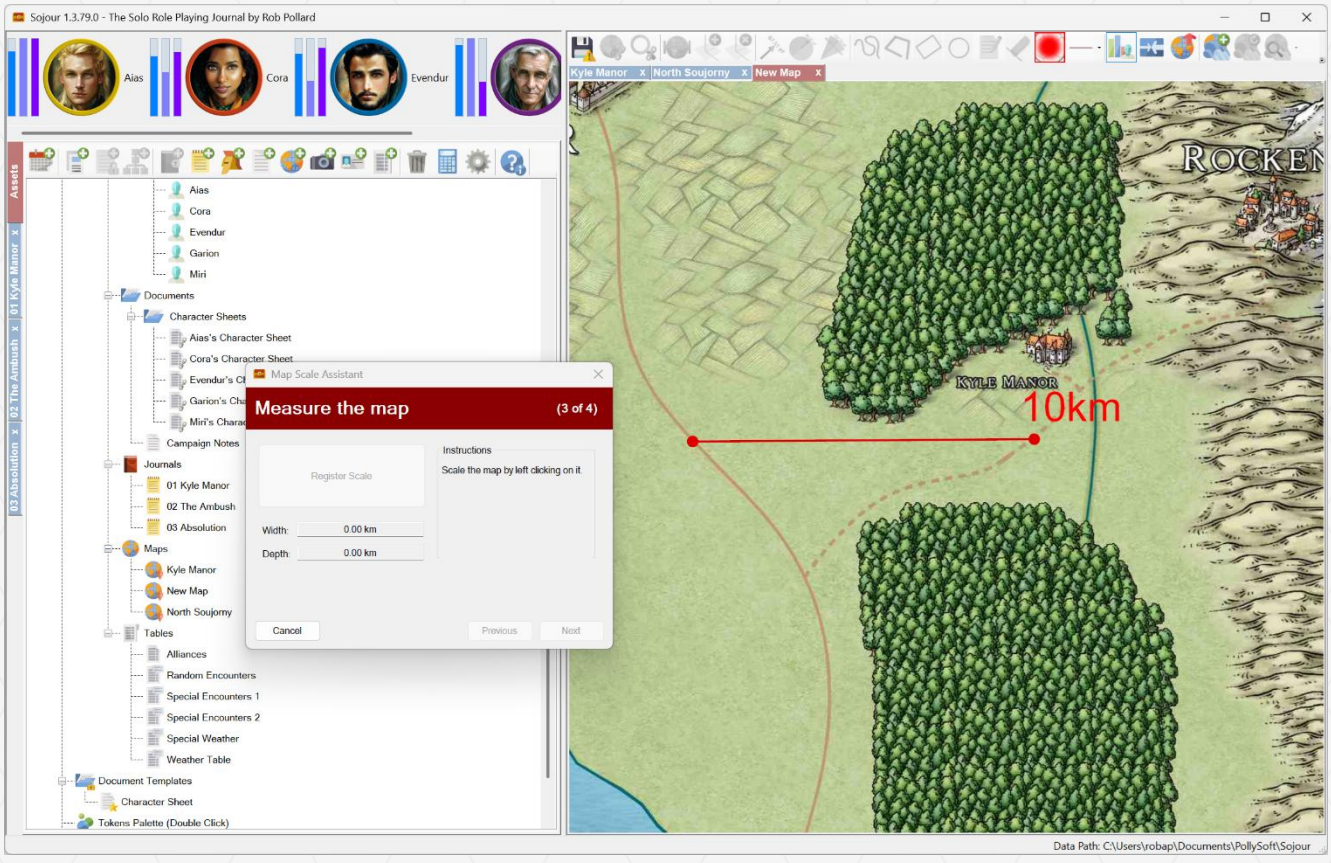
This is the page from where you will initiate your map measurements by clicking the **Register Scale** button. You can attempt to register the map as many times as you wish and can even go back through the wizard to change some earlier settings.

This page's width and depth fields will have the current map's width and depth measurements or 0 if the map is not registered.

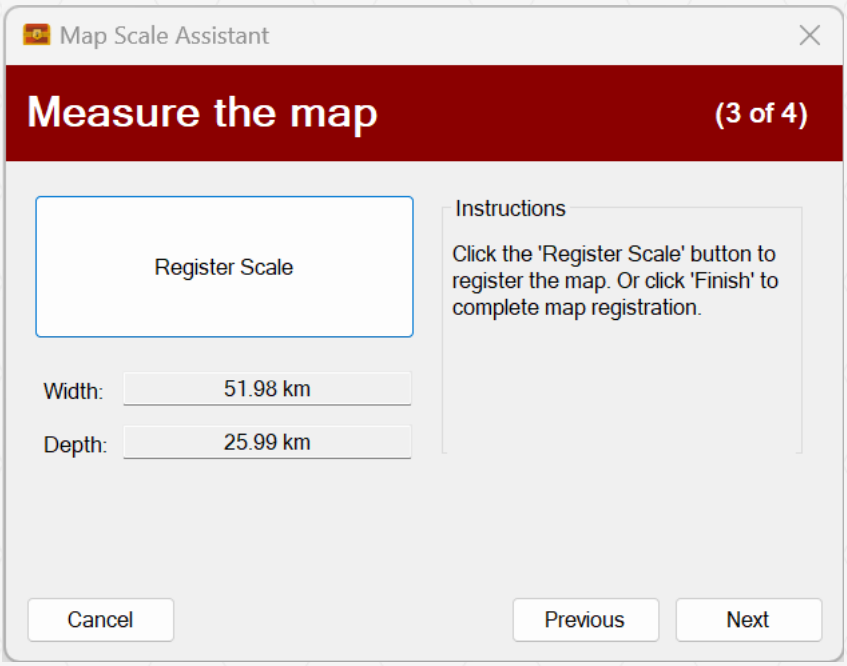
Clicking the **Register Scale** button changes the instructions in the wizard to:



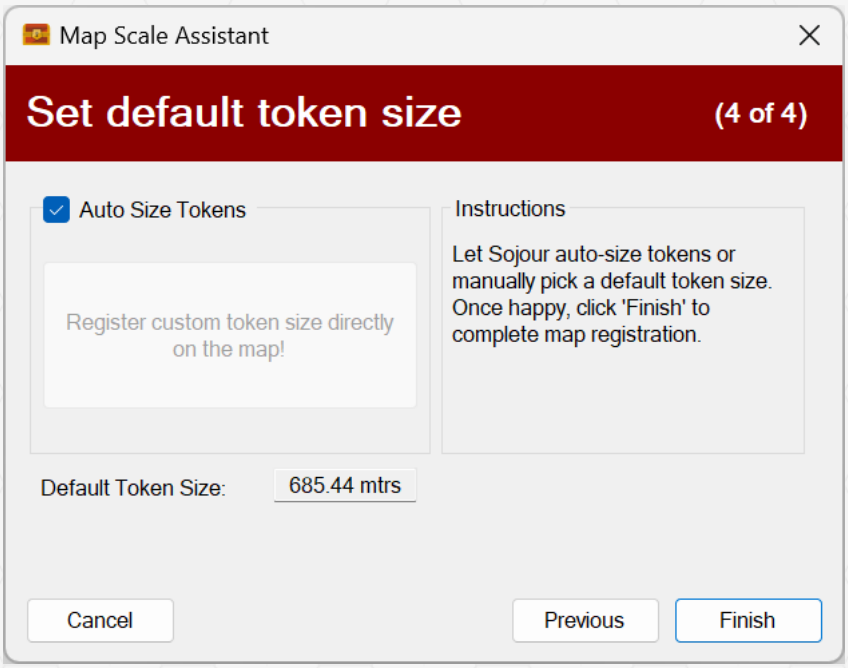
Simply do as it says and left click on the map where you want to start your measurement. The map can be panned and zoomed whilst taking the measurements. Here we are taking our measurements after the first left click:



Once you are happy with the measurement on the map, left click a second time to lock it in. At this point the measurement tool will disappear from the map and the wizard will update to reflect your new measurements:



In the above example the wizard has calculated the map size to be 51.98 km by 25.99 km. If you are happy with these measurements click **Next** to move onto the last screen of the assistant:

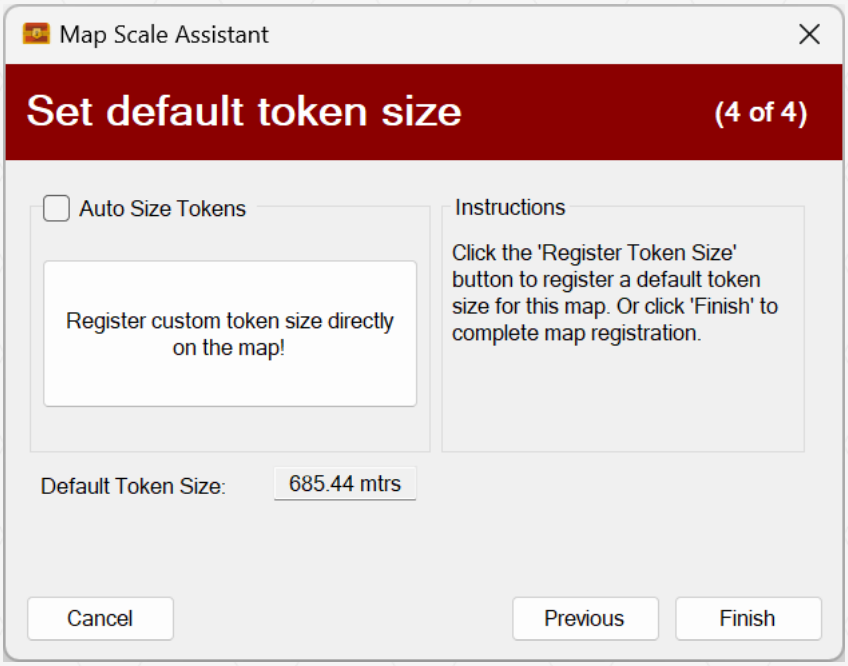


The last page allows you to set a default token size for your map. The size shown is for a token with a relative size of 1. In general, Sojour is pretty good at estimating a good size for your tokens, so if you agree with the displayed size, simply leave **Auto Size Tokens** enabled and click **Finish**.

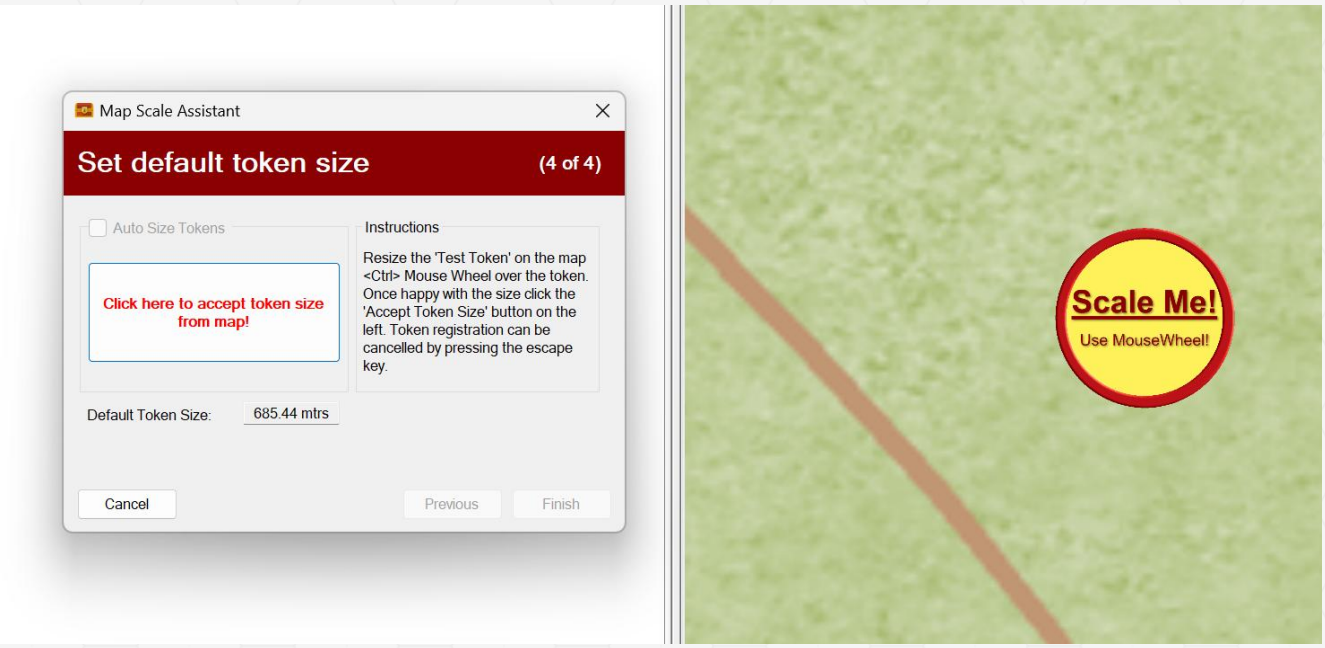
Note, if you are opening the assistant on an already registered map, the **Default Token Size** and the state of **Auto Size Tokens** checkbox will be default to the current map's settings.

The units used to measure the tokens are always based on the units chosen on the first page of the wizard.

There may be occasions where you might have imported a map with hexes or a grid and you might want your tokens to be a very specific size in relation to the hexes or grid. In this case, you should turn off Sojour's auto token sizing by unchecking the **Auto Size Tokens** checkbox:



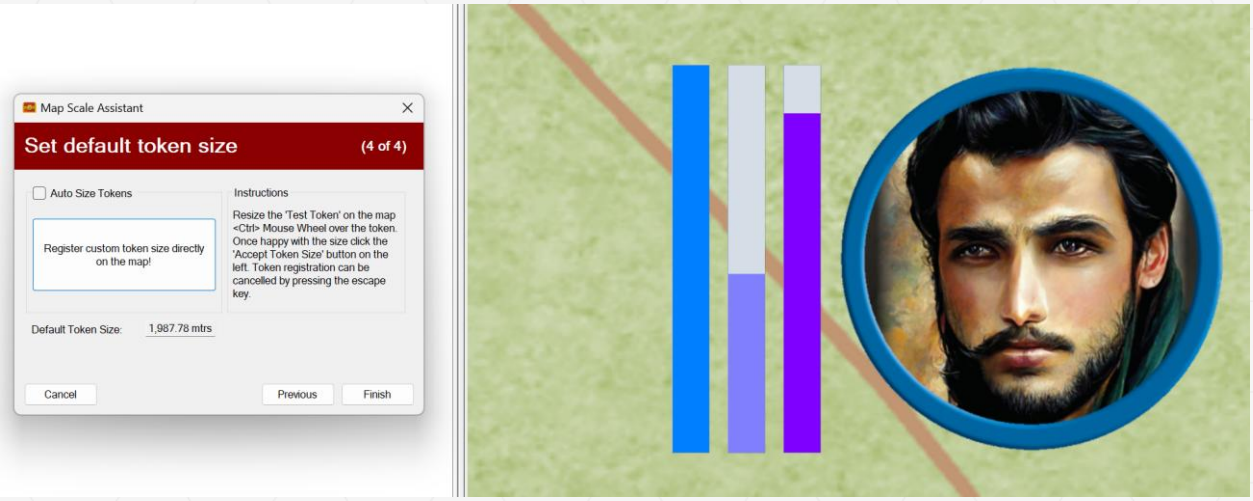
Once unchecked you are offered the option of setting a custom token size for your map. To do so, click the **Register custom token size directly on the map!** button:



This will result in all tokens on the map being hidden and a new yellow one being inserted called **Scale Me!** As shown above.

The **Scale Me!** token can be dragged around the map to position it where you want it and it can be resized by using a mouse wheel with the mouse pointer directly over it:

Once you are happy with the token size, simply click the **Click here to accept token size from map!** button on the assistant:



This results in the scaling token being removed and all original tokens being put back on the map, but this time at the newly chosen size.

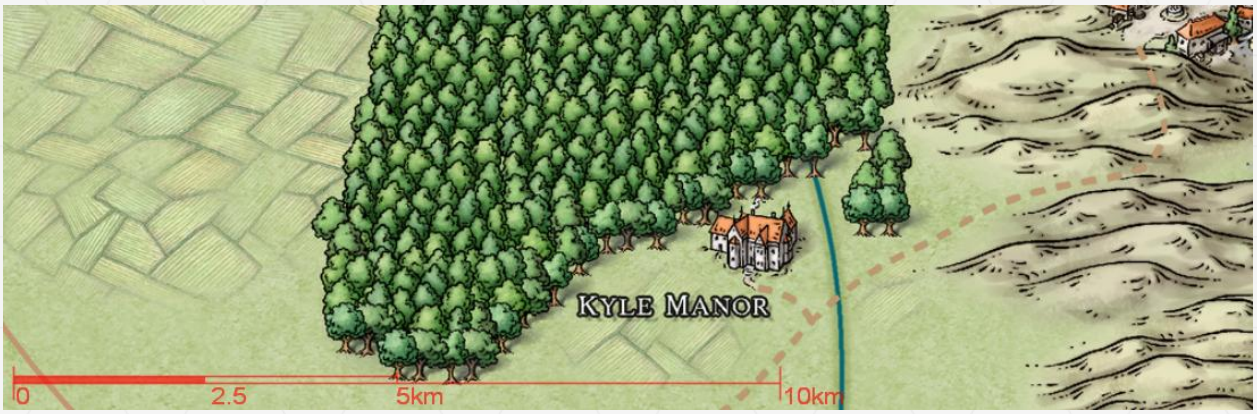
If you don't like the size, simply rescale the token again, or alternatively, re-check the **Auto Size Tokens** checkbox to have Sojour automatically size your tokens for you.

Existing tokens that have been manually scaled with the mouse wheel will be scaled in proportion to their size relative to the standard token size.

Clicking the **Cancel** button at any point will revert all your changes and close the assistant.

When you are ready to complete map registration, simply click **Finish**.

Once you have clicked **Finish**, the map will be scaled which will result in a map scale being displayed on the map:

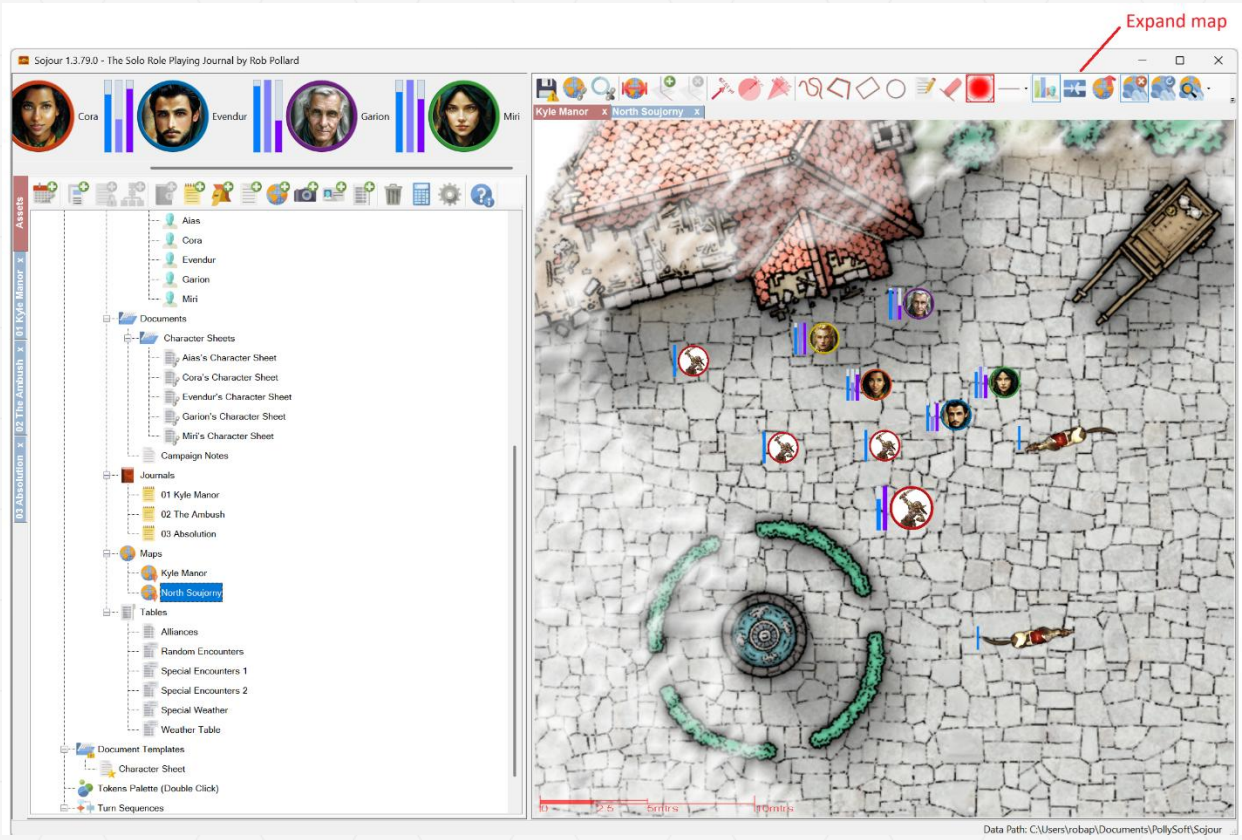


The new scale will adopt the colour chosen during the scaling process. If you are not happy with the colour, rescale the map and pick a new colour.

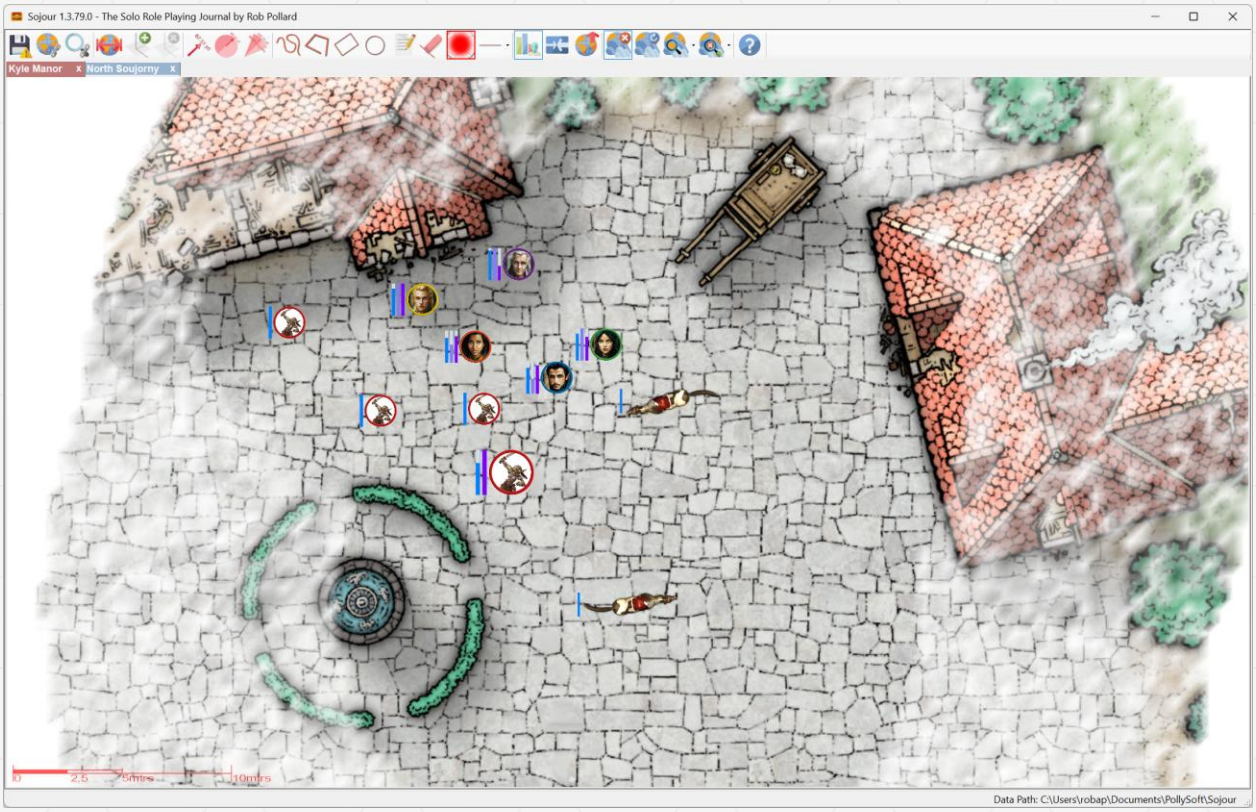
Any tokens already on the map will be resized to match the new map scale.

Expanding maps

Maps can be expanded to fill most of the screen with the **Expand Map** toolbar button:



The first click of **Expand Map** will hide Sojour's left hand pane to enable the map to fill the entire window:



This could be a handy feature for anyone planning to use Sojour as a mapping surface.

When a map is expanded, the **Expand Map** button will be renamed to **Restore Left Pane**. Clicking this button once more will restore Sojour's left hand pane.

Sojour will automatically **Restore Left Pane** when the last opened map has been closed.

Warning! The **Expand Map** function is one of the few user interface functions that is not persisted between instances. Sojour will always start with its left pane visible.

Getting map help

You can obtain a list of all the map shortcut keys by clicking the **Show keyboard shortcuts for maps** button:

Clicking this button will display the following stay-on-top helper window:

Key	Action
<Esc>	Exit all measurement and drawing modes
W	Pan map upwards
S	Pan map downwards
A	Pan map left
D	Pan map right
Q	Zoom in map
Z	Zoom out map
Mouse wheel	Zoom map
Mouse wheel on unlocked token	Resize token
<Ctrl> Mouse wheel token	Rotate token
Left click drag token	Move token with animated journey
<Ctrl> left click drag token	Move token instantly with no animation
<Ctrl + Shift> left click token	Get character or npc to toggle conversation
Right click token	View token's context menu
Double click token	Opens the token's associated document (if any)
<Ctrl> Mouse wheel map link	Resize map link
<Ctrl> Mouse wheel range arc	Change the arc width in degrees
Right click line drawing	Exit line drawing mode and terminate current line
Mouse wheel rectangle	Rotate rectangle
<Ctrl> mouse wheel rectangle	Alter rectangle's width
<Shift> mouse wheel rectangle	Alter rectangle's height
Mouse wheel circle	Alter diameter of circle
Mouse wheel text	Rotates text
<Ctrl> mouse wheel text	Resizes text
<Shift> mouse wheel text	Change text opacity
Mouse wheel healthbar	Raise or lower the npc's first characteristic
<Ctrl> Mouse wheel healthbar	Raise or lower the npc's second characteristic
<Shift> Mouse wheel healthbar	Raise or lower the npc's third characteristic
<Shift> <Ctrl> Mouse wheel healthbar	Raise or lower the npc's forth characteristic
Mouse wheel tilt registration	Change size of cursor - affects visual only
<Shift> mouse wheel tilt registration	Rotate in roll
<Ctrl> mouse wheel tilt registration	Rotate in pitch

Close

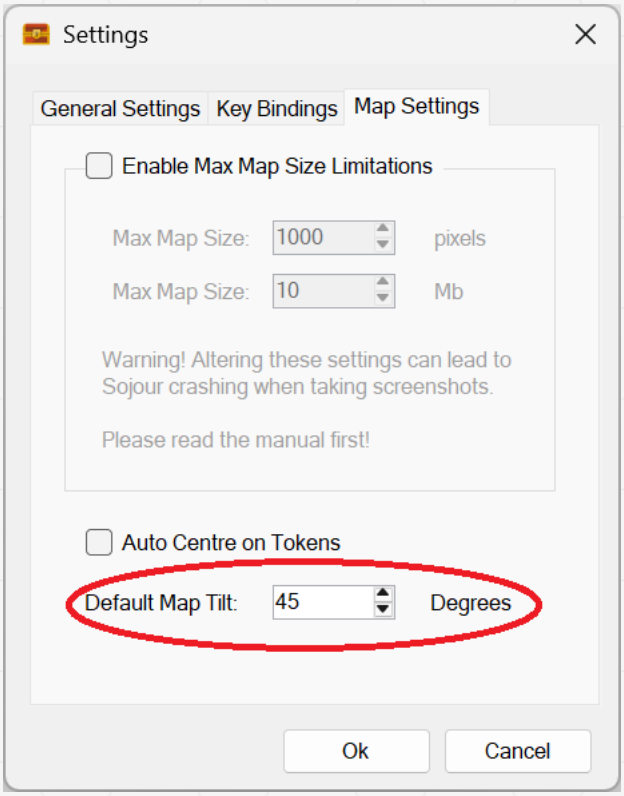
This window shows all keyboard shortcuts related to maps. It can be dragged to any point on the desktop by left clicking on it and dragging the window.

The window is not a modal window, which means you can carry on operating Sojour with it open.

In addition, the window will remember where it was last left, so when re-opened it will appear at that location.

Tilting Maps

Sojour's mapping pane is rendered using Ionian, a custom 3d engine. This allows a user to be able to toggle the tilting of each map by an angle defined under Sojour's map settings:



The default tilt angle is set to 45 degrees, but this can be changed from any angle from 5 degrees to 85 degrees. Changing this setting will affect all tilted maps except those that have custom tilt registrations.

To tilt a map click this button on the mapping toolbar:

Clicking this button will tilt the map by the angle defined in settings:

The button will then turn into a **Tilt Map Up** button to allow the user to remove the tilt from the map.

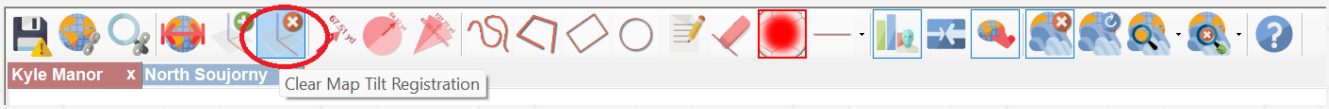
Each map remembers whether it was tilted or not, so they will remain the same the next time Sojour is opened.

Custom Tilt Registrations

In addition to the default tilt angle, a user has the option of setting specific tilt angles for each of their maps. These custom tilt angles are saved along with the map, so that each time it's tilt is toggled on and off it will always tilt to its custom registered angle.

Changing the default map tilt angle in settings has no effect on maps that have custom tilts registered to them.

However, a user can choose to remove a custom tilt from a map by clicking this map toolbar button:

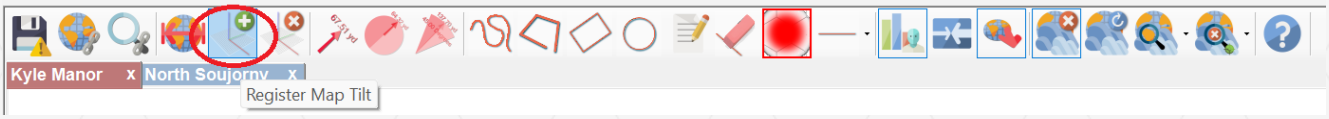


This button will be disabled if a map has no custom tilt registered against it.

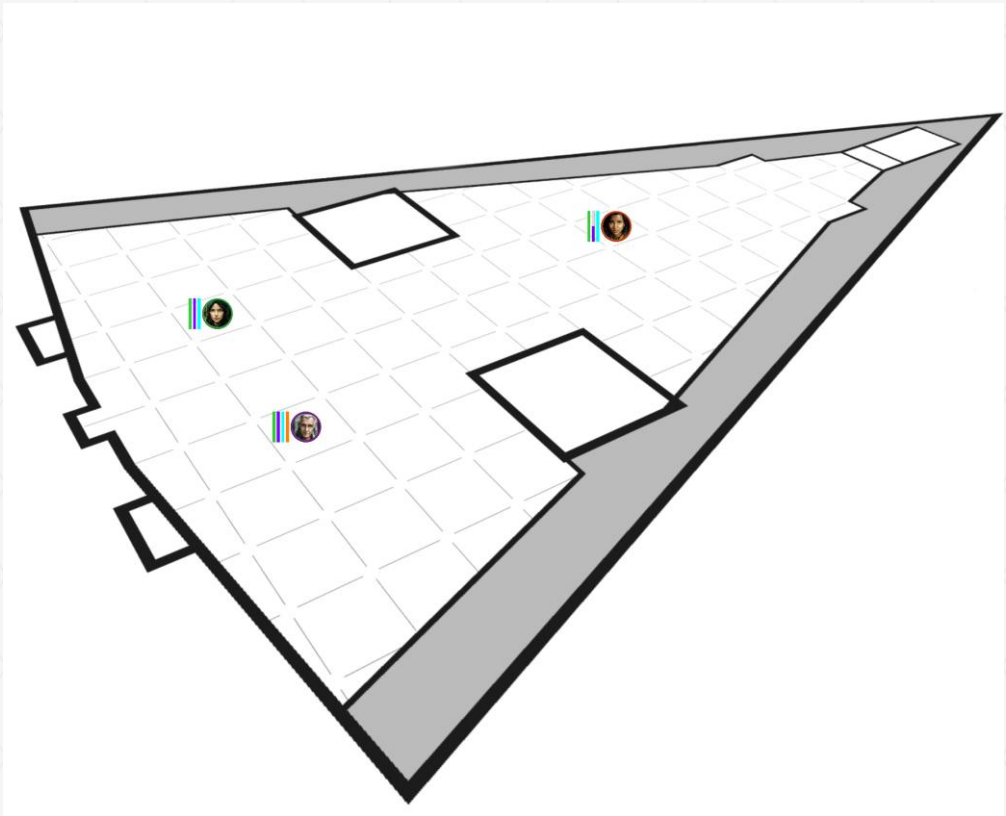
Custom tilt registrations allow the user to precisely set map tilts for each map if the default map tilt doesn't work for a particular map.

The registration tool also allows one to accurately tilt a 2d isometric map so that it is tilted into the exact plane that the artist drew the map. This leads to an uncanny 3d effect for these type of maps!

To register a custom tilt for a map click this map toolbar button:

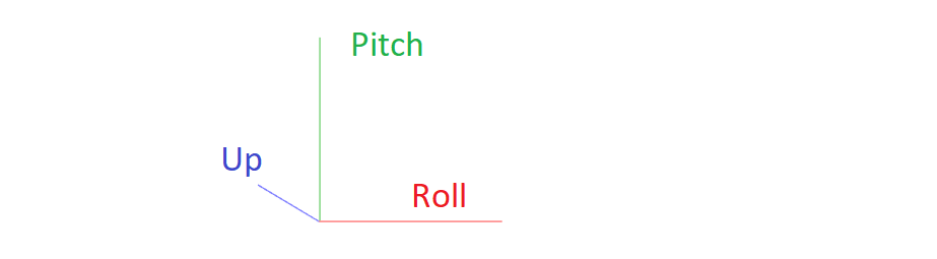


Consider the map below. This is a 2d isometric map. When tokens are dragged to it, the map looks like this:



This is a bare example because I'm not very good at art, so the effects won't be as pronounced as you would see on a professional map.

To tilt the above map, we register a tilt that aligns with the drawing. This displays the 3d tilt cursor:

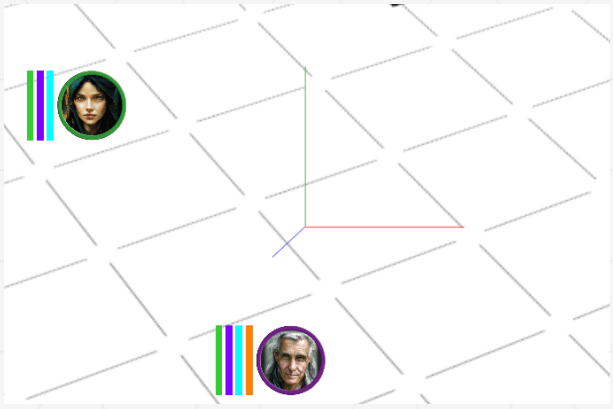


This cursor can be moved around with the mouse. The mouse wheel has three functions that affect this cursor:

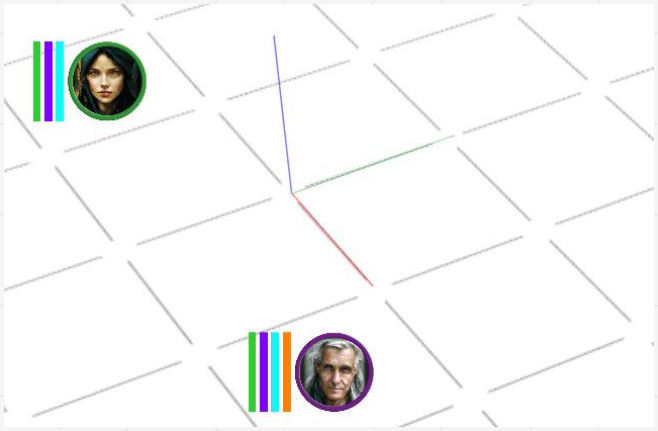
Key Modifier	Result
<None> + Mouse Wheel	Changes the size of the cursor. This has no effect on the final result. The adjustment is provided to help with map alignment.
<Left Shift> + Mouse Wheel	Rotates the **up** axis.
<Left Ctrl> + Mouse Wheel	Rotates the **roll** axis.

The idea is to align the green and red lines with your map's grid whilst also ensuring that **Up** is pointed upwards.

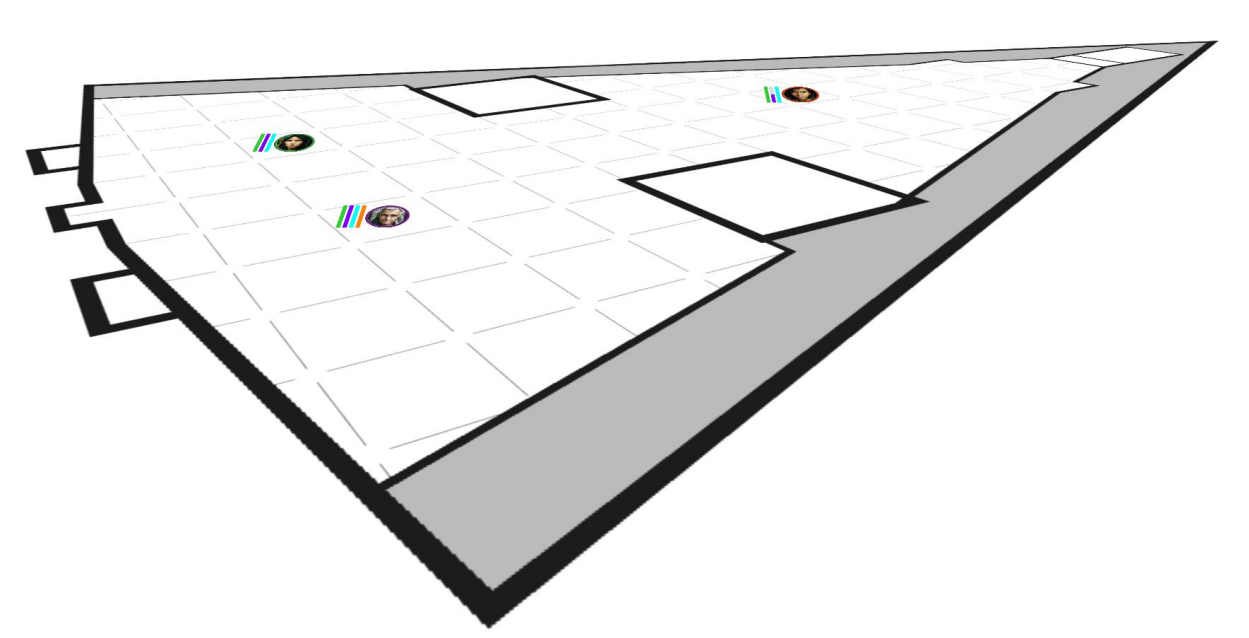
For our example map the cursor initially looked like this:



But with a little adjustment we can get it to look like this – perfectly aligned!:



Once properly aligned click the left mouse button and the map will tilt to precisely match the map's grid:



With good 2d isometric art – this isn't – the effect can be quite stunning. The 2d drawn features tend to 'pop' almost as if they were 3d!

Each map will remember its own custom tilt and that tilt can be toggled on or off at any time.

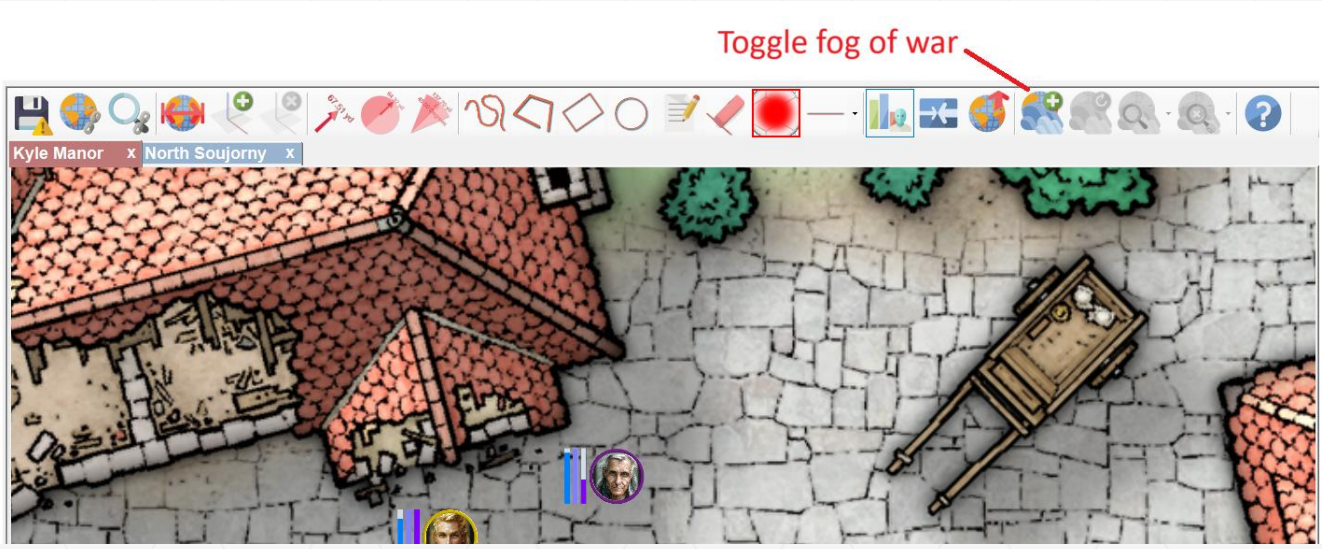
Tips for alignment:

1. The registration cursor length makes no difference. What does make a difference is how parallel the red and blue lines are in relation to the map's grid. The more parallel, the more accurate the tilt!
2. Ensure the **blue up** line is pointed upwards or you will get an upside down tilt! (If you don't like the tilt, either clear it, or re-register it)
3. The pitch and roll axis using the shift and control keys with the mouse wheel push and pull on each other. As you adjust one it will also alter the other. Don't worry! The idea is to iterate between the two axis until the alignment cursor perfectly matches your map as shown above!

Map Fog of War

Sojour maps support Fog of War. Fog of War provides a means of hiding a map in its entirety and only show the parts of the map that tokens have visited. Static campaign asset tokens are treated as part of the map, so they never clear the fog.

Fog can be toggled on and off at any time with this button:



Here is the same map after Fog of War has been toggled on:

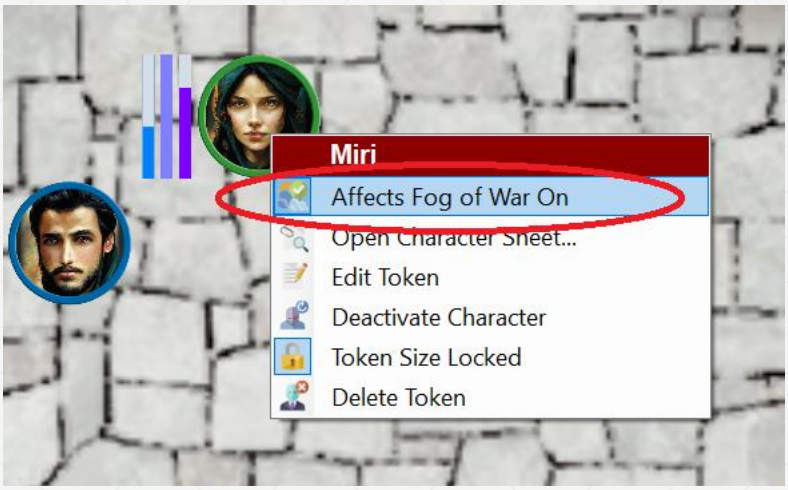
Sojour has automatically hidden the map in Fog except where it has been revealed by tokens.

Tokens can be toggled to affect Fog of War or not. Those that don't, never affect the fog. Those that do, clear the fog around them.

Sojour defaults whether a token can affect Fog of War based on the token type:

Token Type	Default Affects Fog of War?
Character	Yes
Token from token palette	No
Static Campaign Asset	No
Non Static Campaign Asset	Yes

All of the above token types can have its **Affects Fog of War** attribute toggled on or off by right clicking on the token and clicking on the option shown below:

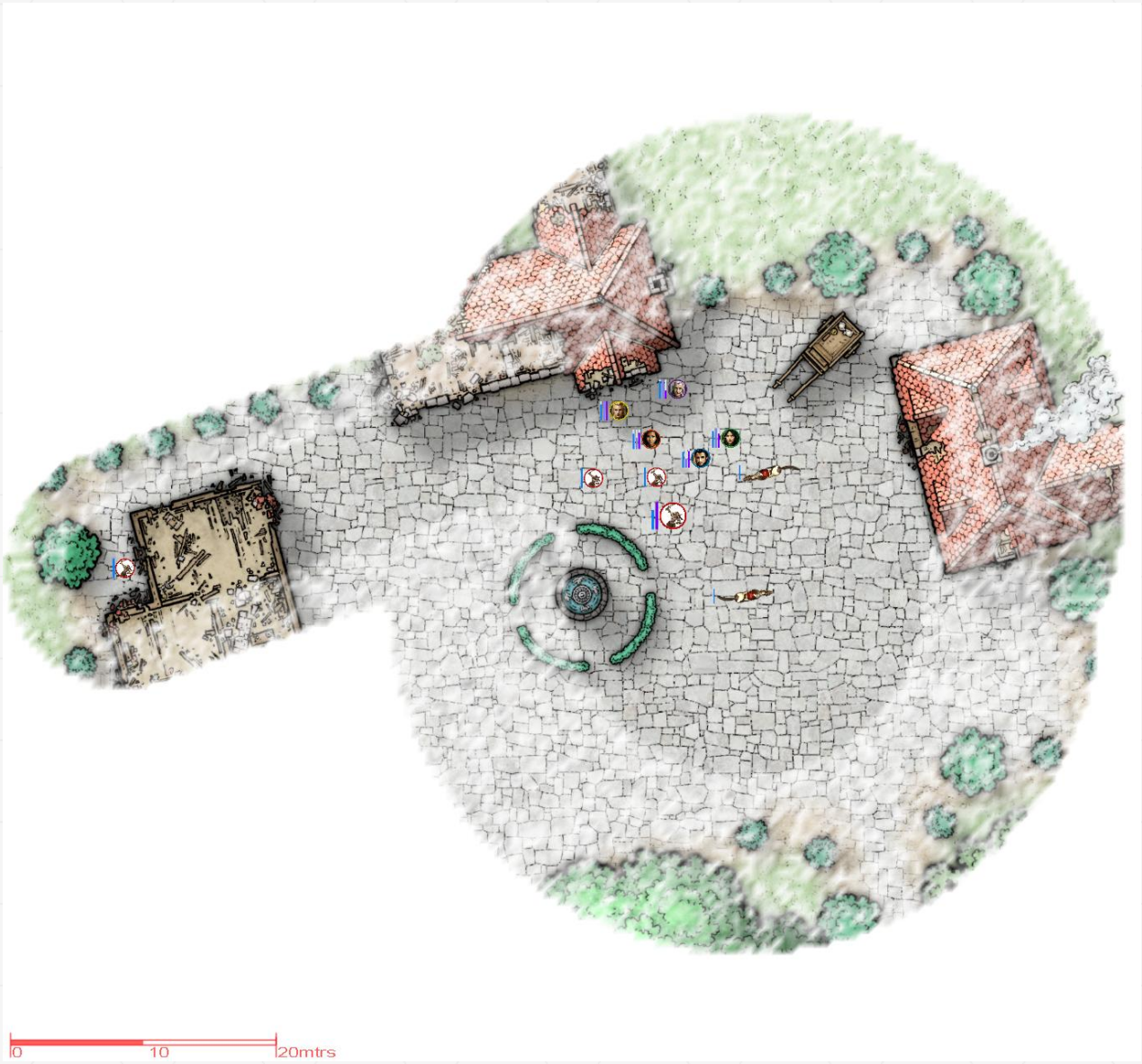


This menu item is only shown when Fog of War is active for the current map.

If a map's Fog of War is on and a token is dragged to the map from the Token Palette, Sojour will automatically switch on the mouse preview to allow you to see enough of the map to allow to decide where to drop the token.

Once the token is dropped, Sojour will revert the mouse preview back to its original settings, which is usually off.

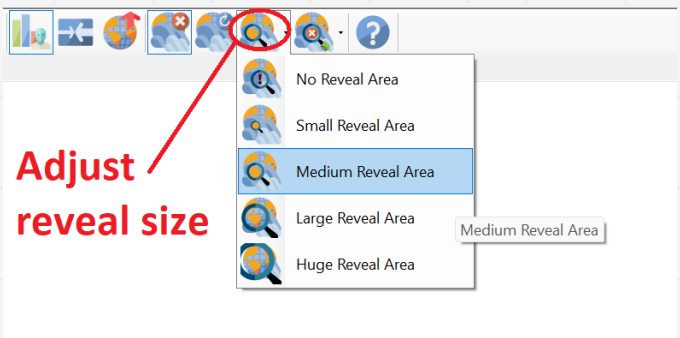
Moving a token or character will cause more of the map to be revealed:



Here we can see that a goblin top right has decided to explorer to the West!

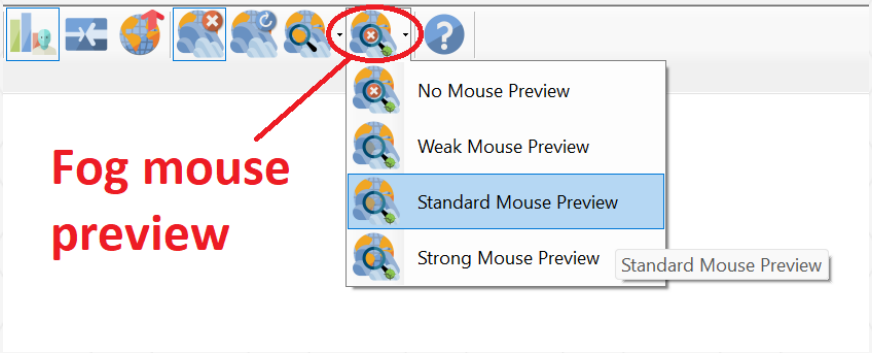
The Fog calculations do not use line of sight. Instead they use token widths. The wider the token the bigger the rift created through the Fog.

In addition Sojour provides five different reveal sizes ranging from none to huge:



Sojour defaults to Medium Reveal Area and this is shown in the preceding screenshots. These settings, like all the Fog of War settings are per-map and are stored with that map. So when Sojour restores each map will have its own custom Fog of War settings.

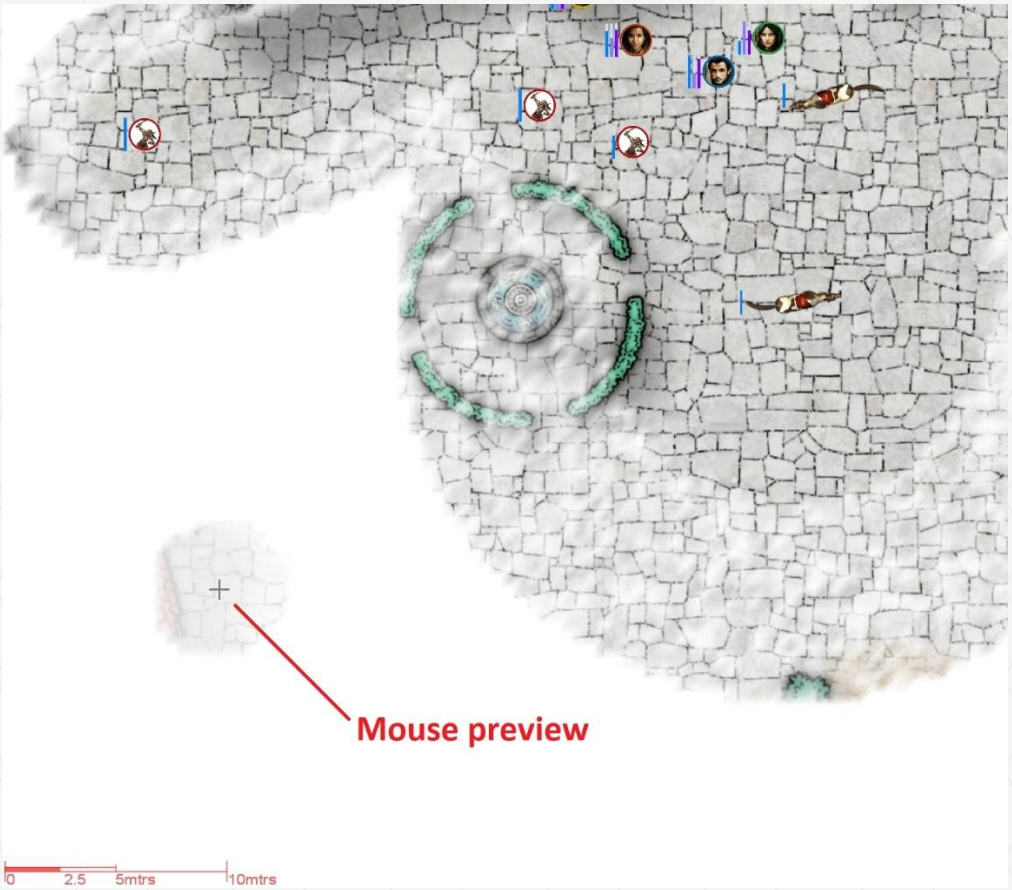
Sojour also incorporates a tool that lets you use the mouse cursor to see through the fog. The tool is activated with this menu button/drop down:



The mouse preview defaults to off, but you can set a variety of intensities, with weak only just peering through the fog and strong piercing right through it. This window through the fog will follow the mouse cursor around.

Warning! Alas, due to technical limitations, the mouse preview will not work whilst dragging tokens. There is a story in the system to actively work on a resolution for this. But right now it's how its designed.

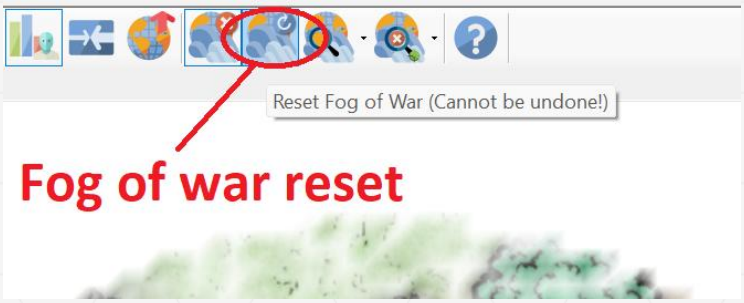
Here is the **Standard Mouse Preview** in action:



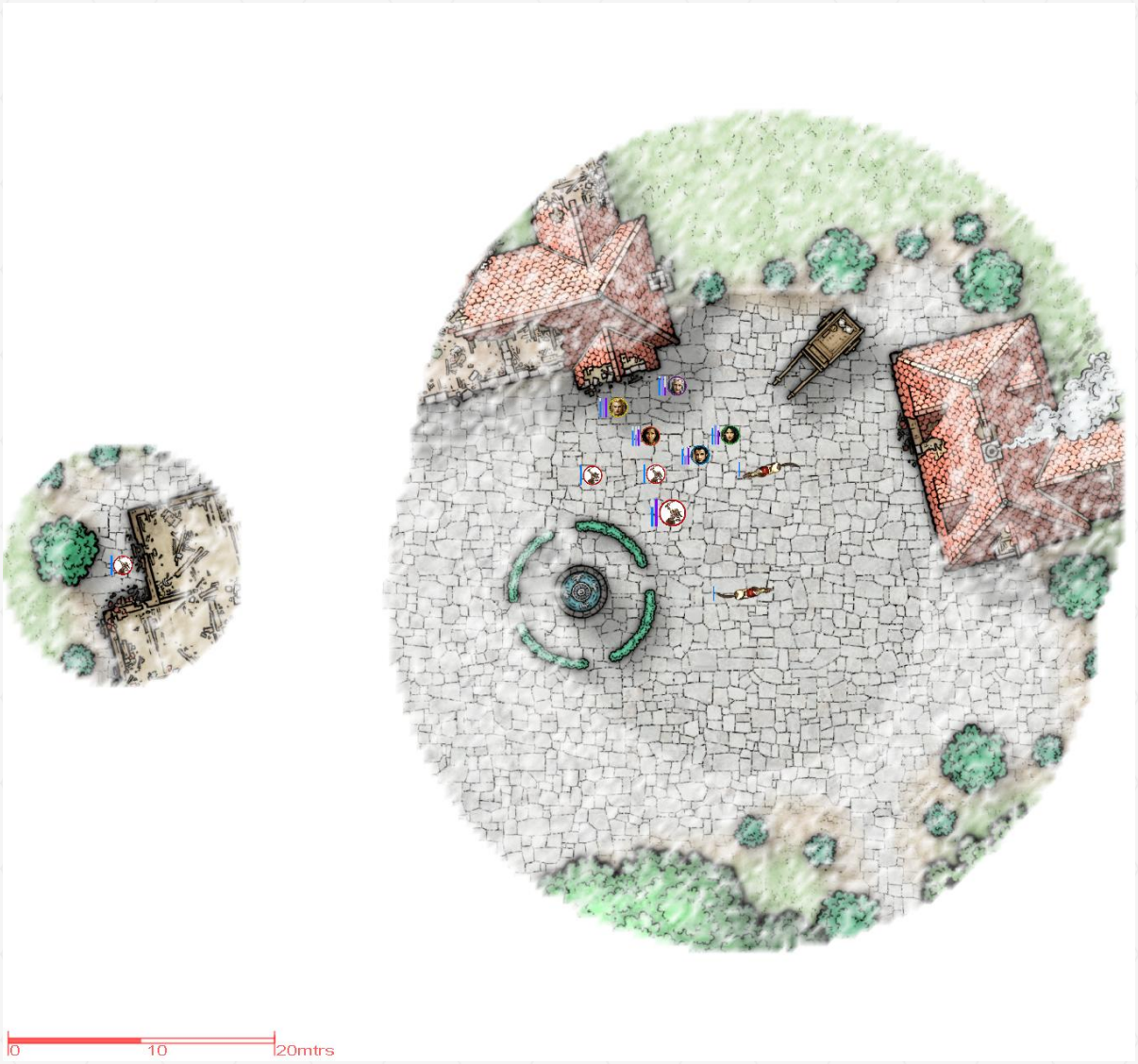
We can see that it has made a temporary hole in the fog that will follow the mouse cursor around!

The preview window size is set by the reveal size drop down.

Fog of War can be reset at any time by clicking the reset button:



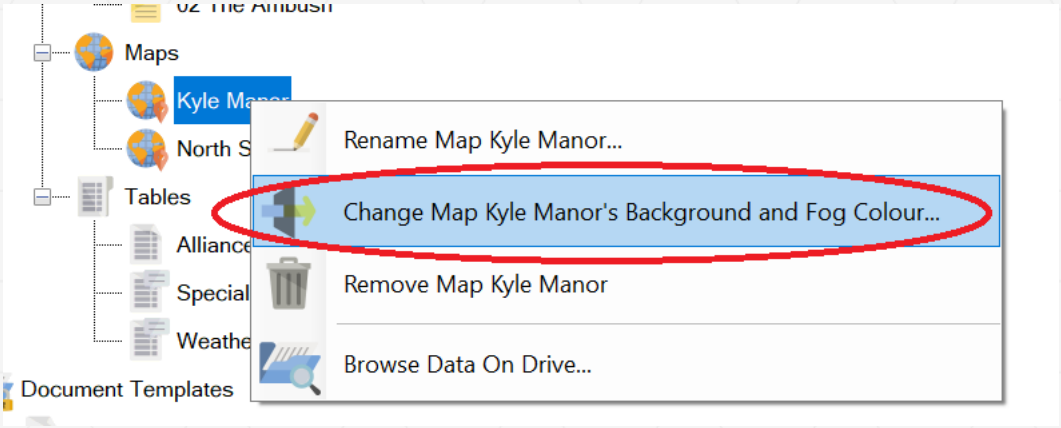
Clicking this button will erase all discoveries and will revert to what the current tokens can currently see:



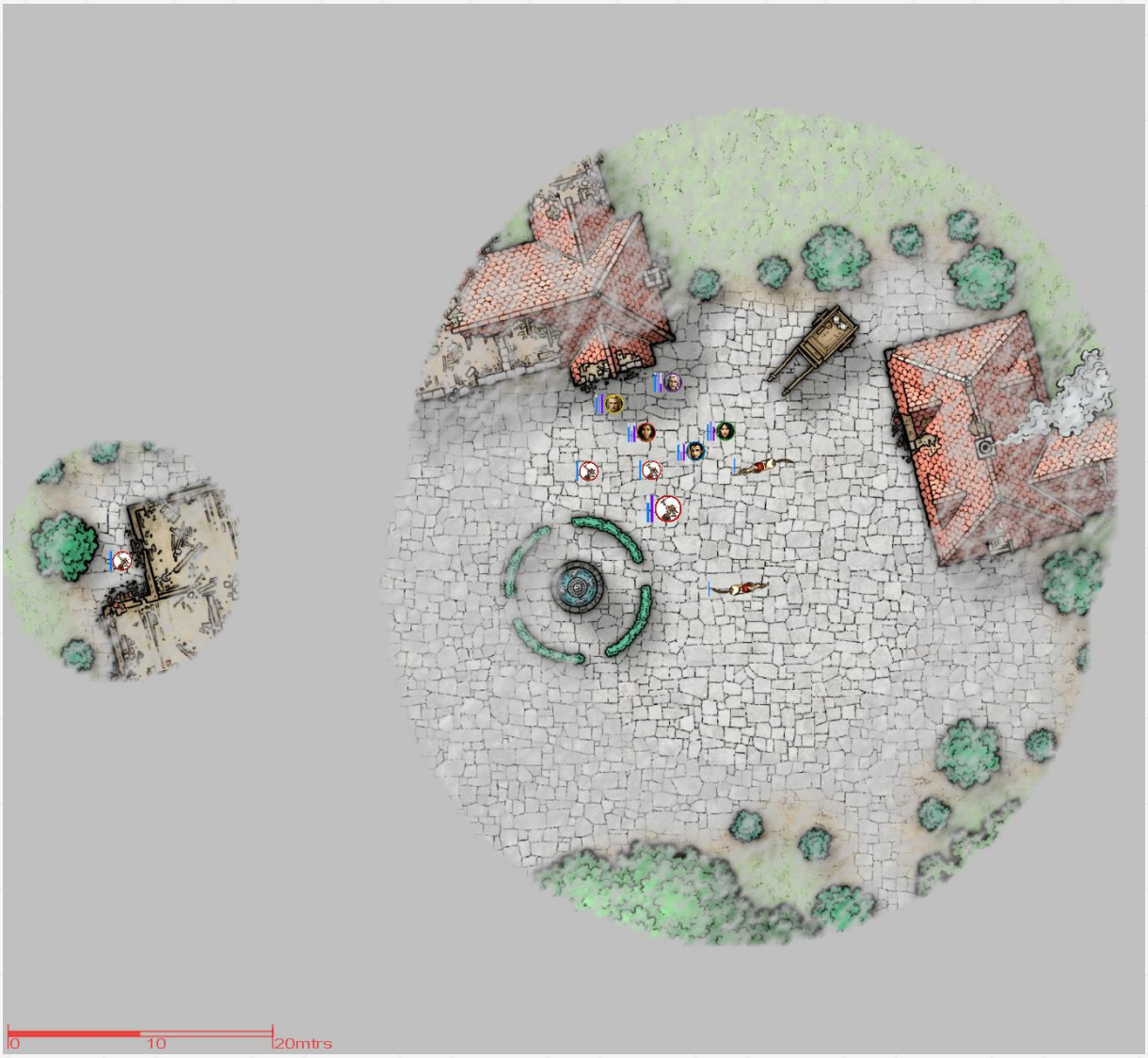
There is no undo feature, so be careful about using the reset button!

Sojour's fog has the same colour as the background colour set for the map when it was first imported.

This background colour can be changed at any time by right clicking the map in the **Assets Browser** and selecting **Change <Map Name>'s Background and Fog Colour…** as shown below:



Once clicked the user will be presented with the standard Windows colour picker. For this example we decide to go for a grey fog and this is the result:



The colour can be set to any colour that you want and each map will remember its own colour settings.

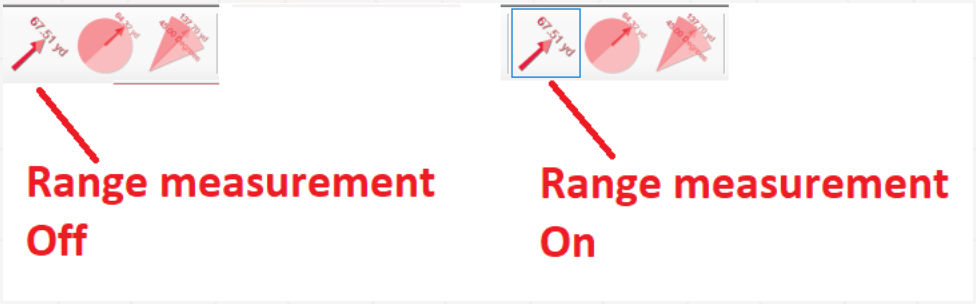
Finally, maps that have Fog of War turned on can have it turned off with this button:

This will revert the map back to no fog. However, the map still remembers previous discoveries, so if the Fog of War is turned back on again, all your previous discoveries will be maintained. You will need to use the Fog of War Reset button if you wat to reset your map's discoveries.

Map Measurement Tools

Sojour provides a number of tools to help you take measurements from your maps. Only one measurement tool can be active at any one time. Picking a different measurement tool will close the previous measurement tool, as will selecting any of Sojour's map drawing modes.

All of the measurement tools can be toggled on or off by clicking their toolbar button. When a tool is toggled on, its toolbar button will be surrounded by a blue outline as shown bottom right:



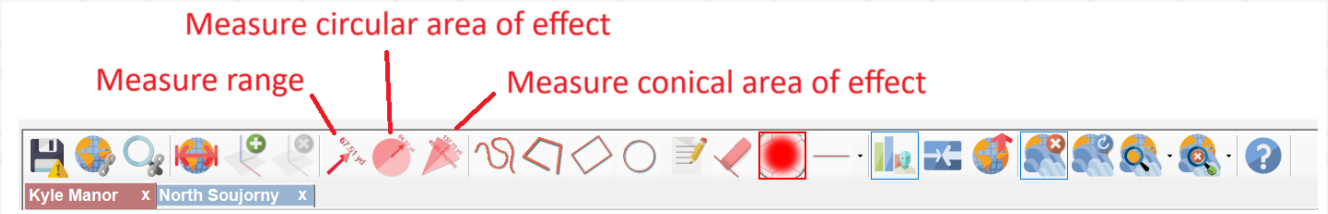
You can continue taking measurements with the selected tool until it is toggled off. A measurement tool can be turned off in one of three ways:

1. By clicking its toolbar button once more to toggle it so that the blue outline disappears.
2. Pressing the <esc> key.
3. By selecting a different measurement or drawing tool.

Warning! Distance measurements and the measurement tools will be disabled if the map hasn't been scaled. This is why I recommend that most maps should be scaled, especially if they are to be used for tactical combat.

All measurements are made in the measurements system of the ruleset that the map belongs to.

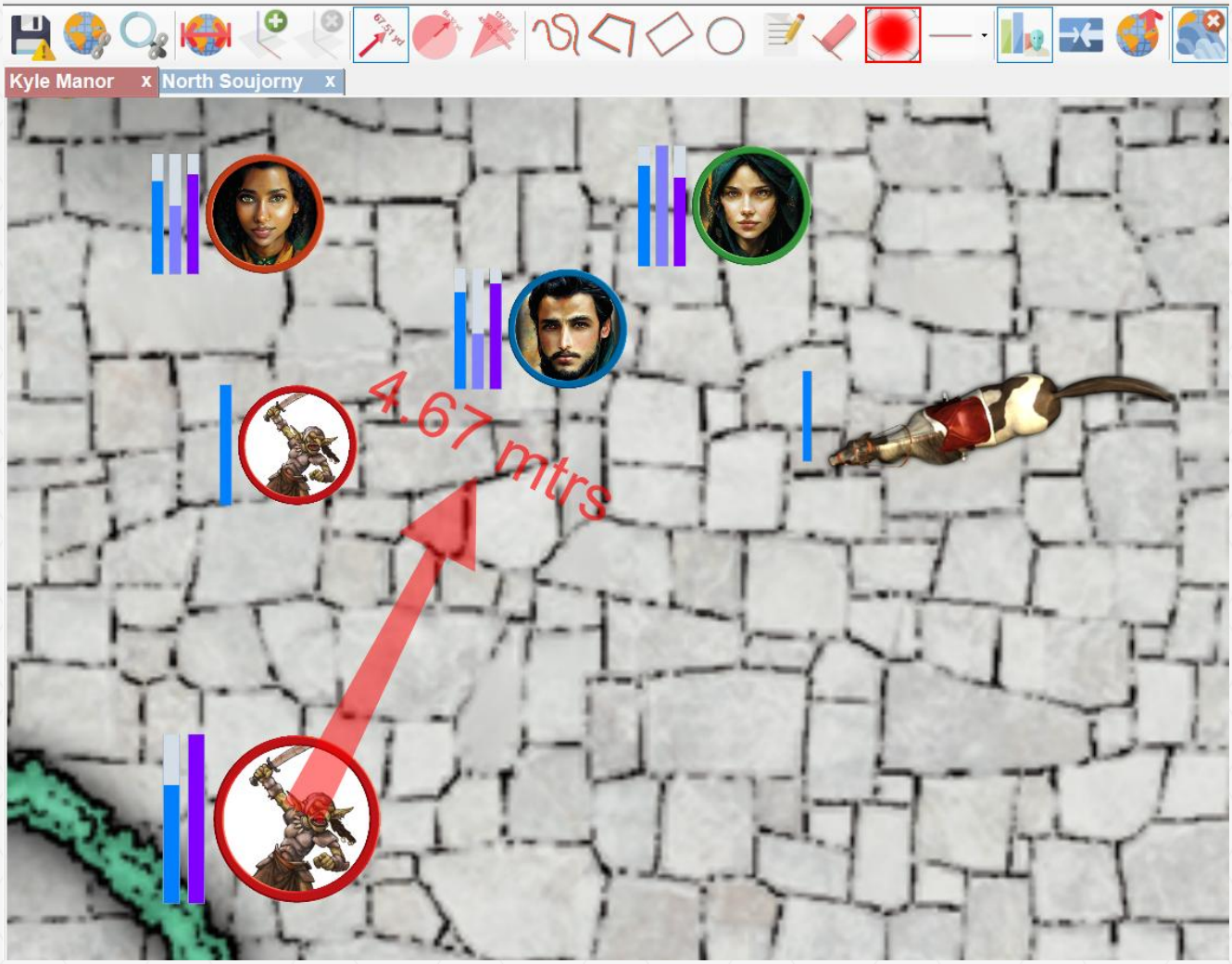
Sojour provides three map measurement tools. Each is activated by clicking one of the tool buttons shown below:



All of the following examples will be displaying the measurements using the metric system. However, Sojour also supports the Imperial System of measurement too. The reason for showing metric measurements in the examples, is due to the maps being located under the Window Guardians ruleset which is defined as using the metric system.

Always take care to pick the correct measurement system when creating your rulesets!

Distance measurement



To measure a distance on the map, click the **Measure Range** button (the blue outlined button in the image above). Then left click the mouse once on the map to anchor the start of the range arrow. Left click a second time to anchor the other end of the range arrow to the map.

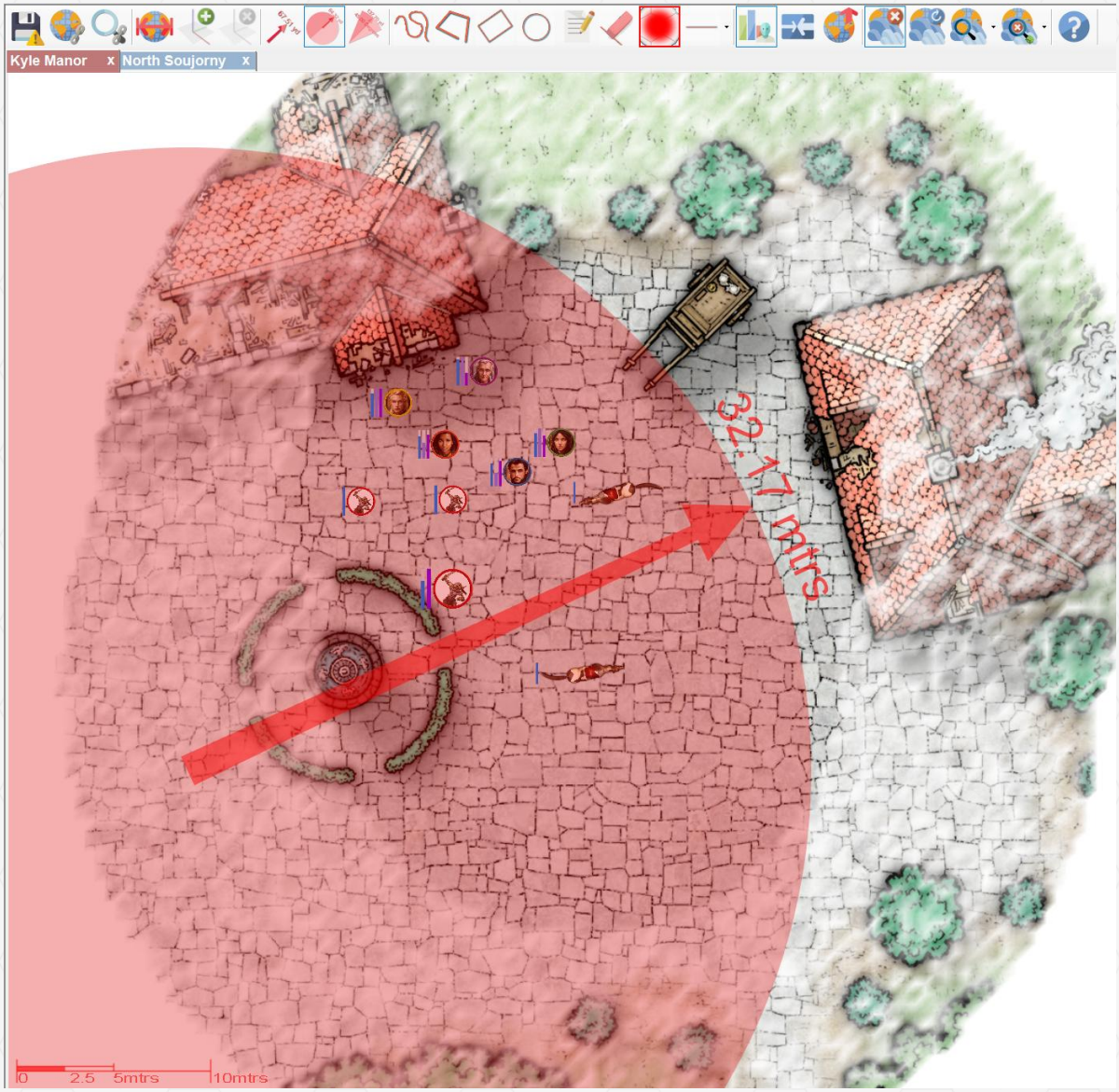
Once a range arrow is fully anchored to the map, it becomes part of the map, so that it will pan and zoom with the map.zzz

To take another range measurement, left click the mouse again at the point where you want to start your new measurement and continue as above.

You can pan and zoom the map while taking measurements. Sojour will auto-scale the measurement tool appropriately when this is done.

Click the **Measure Range** button once more on the map toolbar to exit range measurement mode and remove any range arrows from the map. Pressing the <esc> key will also exit this measurement mode.

Circular area of effect measurement



To measure a circular area of effect, click the **Range Circle** button on the map's toolbar (the blue outlined one in the image above).

The first left mouse click determines the range circle's centre. The second left mouse click determines its radius.

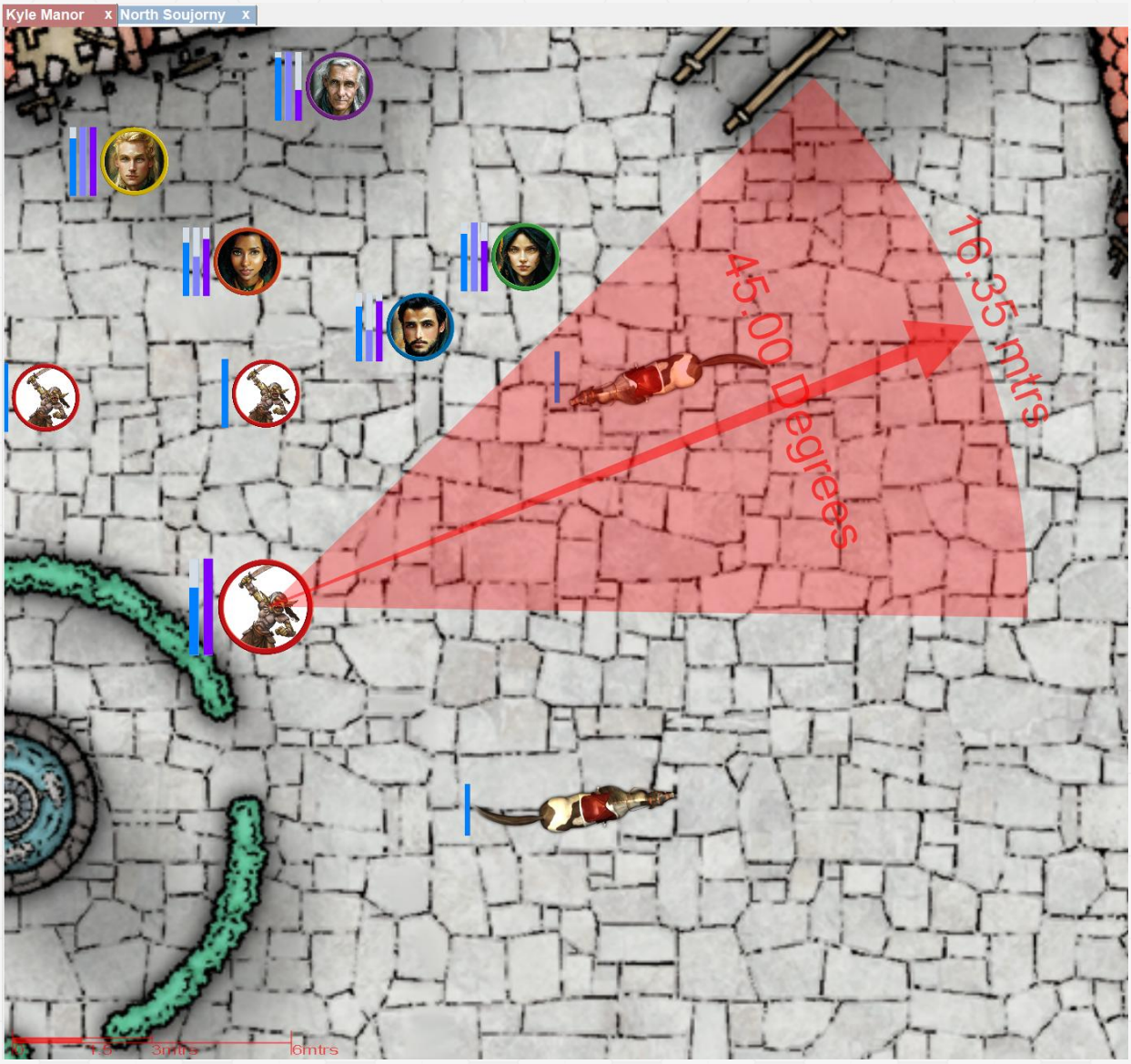
Once the radius has been anchored, the circular area of effect becomes part of the map and will pan and zoom with the map.

To take another range circle measurement, left click the mouse once more at the point where you want the new range circle centre to be, then continue as above.

You can pan and zoom the map when taking measurements. Sojour will auto-scale the measurement tool appropriately when this is done.

Click the **Range Circle** button once more on the map toolbar to exit range measurement mode and remove any range circles from the map. Pressing the <esc> key will also exit this measurement mode.

Conical area of effect measurement



To measure a conical area of effect, click the **Range Arc** button on the map's toolbar (the blue outlined one in the image above).

The first left mouse click determines the range arc's centre. The second left mouse click determines its radius. The width of the arc can be altered by using <Ctrl> mouse wheel.

Once the radius has been anchored, the range arc becomes part of the map and will pan and zoom with the map.

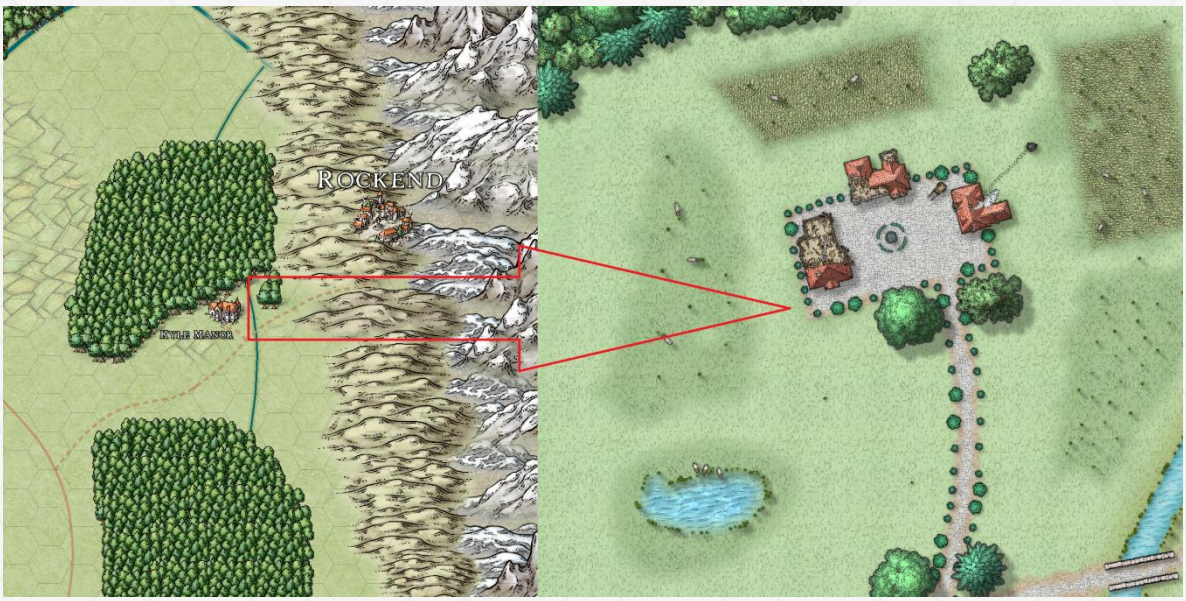
To take another range arc measurement, left click the mouse again at the point where you want the new range arc centre to be, then continuing as above.

You can pan and zoom the map when taking measurements. Sojour will auto-scale the measurement tool appropriately when this is done.

Click the **Range Arc** button once more on the map toolbar to exit range measurement mode and remove any range arcs from the map. Pressing the <esc> key will also exit this measurement mode.

Map Links

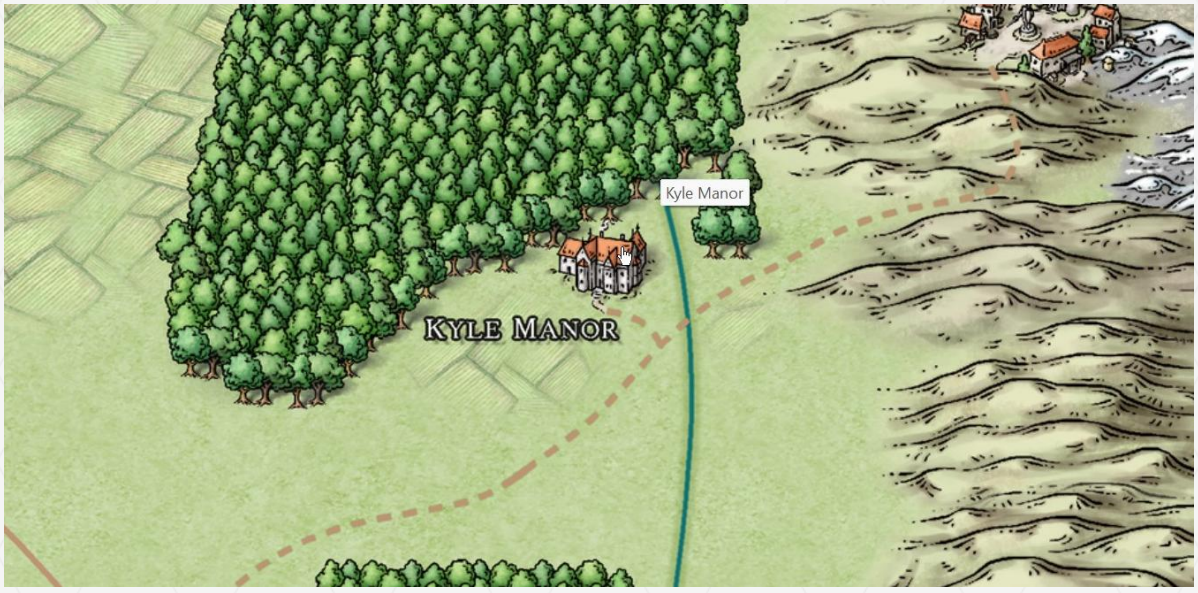
Map links provide a way to link related maps together. Consider the example below:



Here we have two maps. A map of Sojourn on the left and a map of Kyle Manor on the right.

Both of these maps are related. Kyle Manor appears in the centre of the Sojourn map. The Kyle Manor map on the right is simply a zoomed in map showing the Manor in more detail.

Sojour lets the user link related maps together. It does this by creating named circular areas on a map that can be double clicked to open the related map. These circular areas are normally hidden. Their presence is betrayed by a tool-tip being displayed when the mouse cursor is hovered over the circular area:

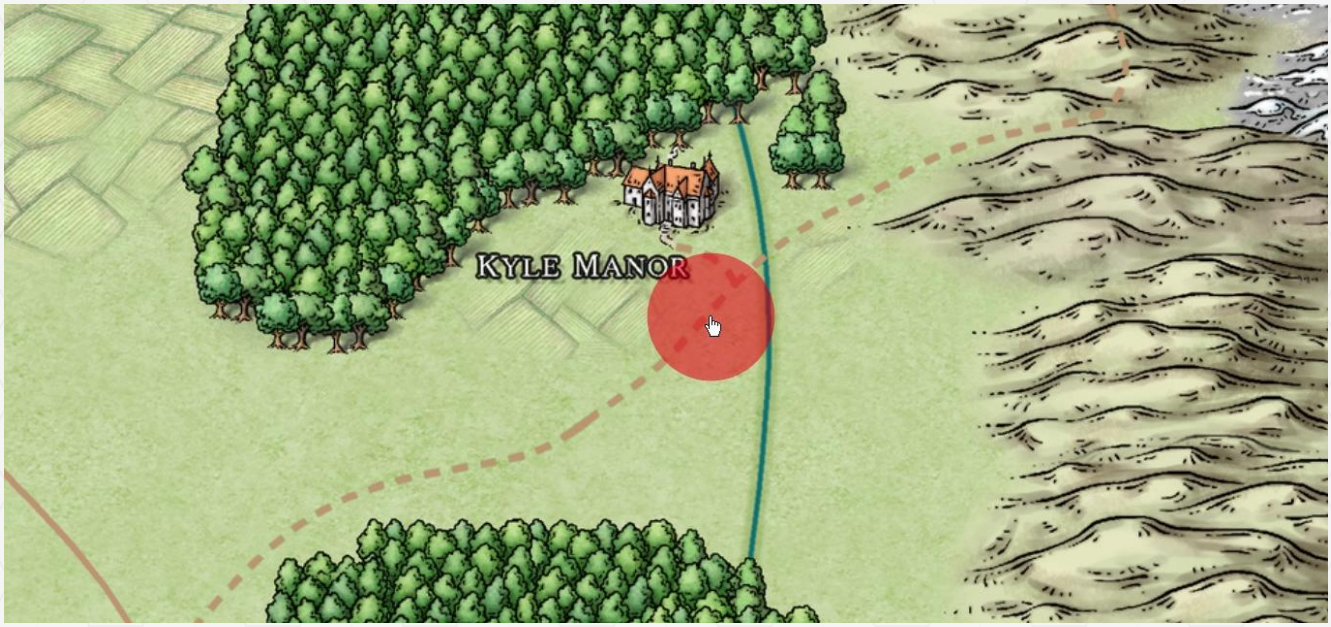


In the above screenshot the user has moved the mouse cursor over **Kyle Manor**. **Kyle Manor** has been registered with Sojour as a map link, this is why the tool-tip appears with the name of the linked map. Double clicking this area of the map will open the **Kyle Manor** map if not already opened. Otherwise, Sojour will simply switch to the **Kyle Manor** map tab.

Adding a map link

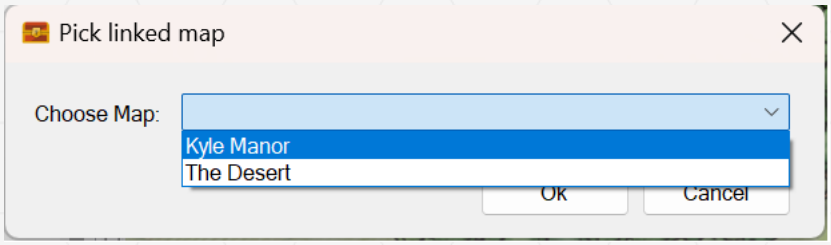
A user can add a map link to a map by right clicking on the map and using the **Add Map Link** menu item, or by clicking the **Add new map link** button on the map's main toolbar:

Once the user has chosen to add a map link, a red highlight circle will become attached to the mouse cursor which is now shaped like a hand:



This area represents the area of the new map link. It can be moved around the map with the mouse and it can be resized using <Ctrl> mouse wheel.

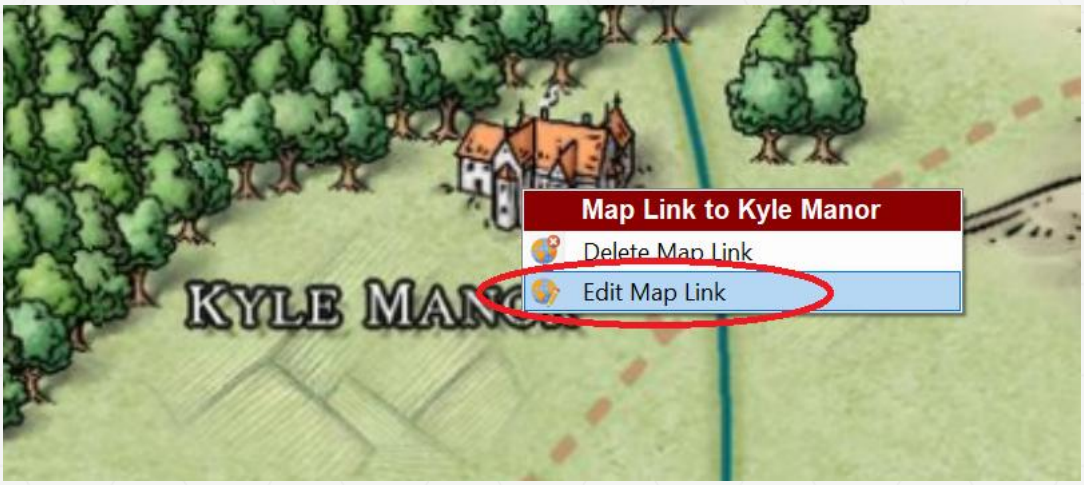
Once positioned and sized where needed, left click the mouse button to anchor it to the map. This results in the **Pick linked map** dialog window being displayed:



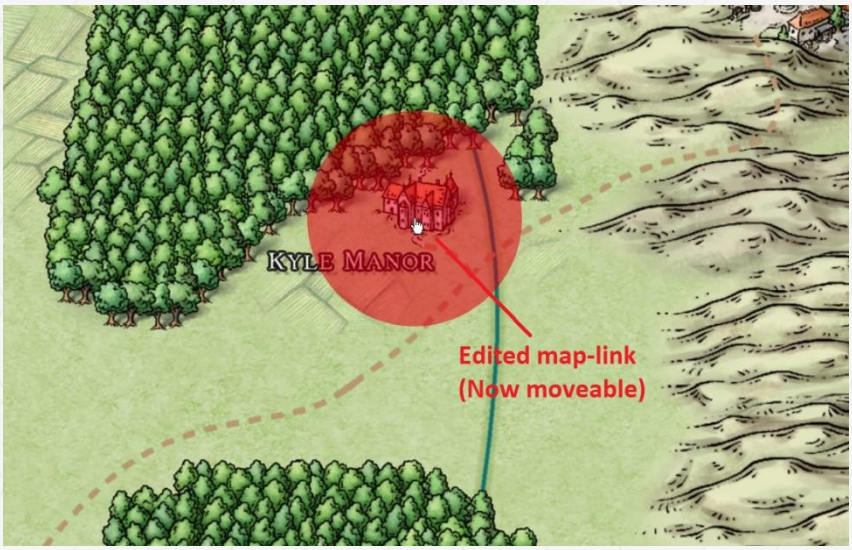
The dialog window will show a list of the other maps in the current campaign. Simply select the map you wish to link to then click ok. You will now have a new map link!

Editing a map link

Map links can be edited by right clicking on them and selecting **Edit Map Link** as shown below:

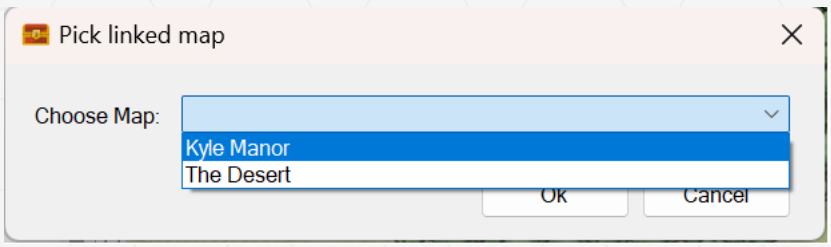


Editing a map link will show the map link area in red and attach it to the mouse cursor, just as when adding a new map link:



The map link can be moved with the mouse cursor. It can also be resized using <Ctrl> mouse wheel and it can be anchored to the map by clicking the left mouse button.

Once anchored the standard **Pick linked map** dialog window will appear:

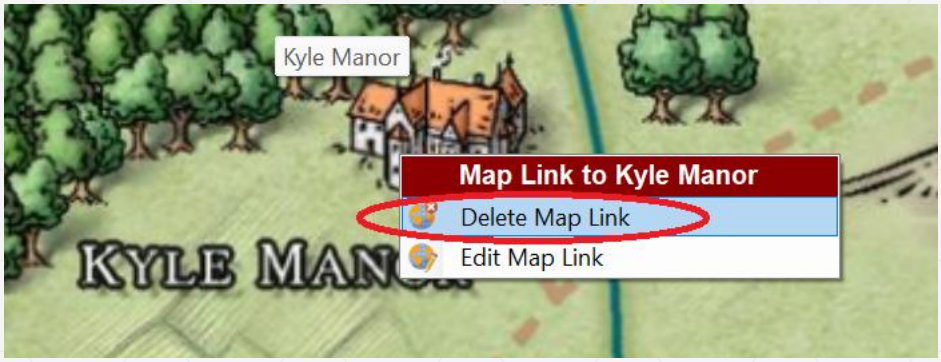


This window will default to the map link's existing linked mapped, but this can be changed if required.

Click Ok to finish editing the map link.

Deleting a map link

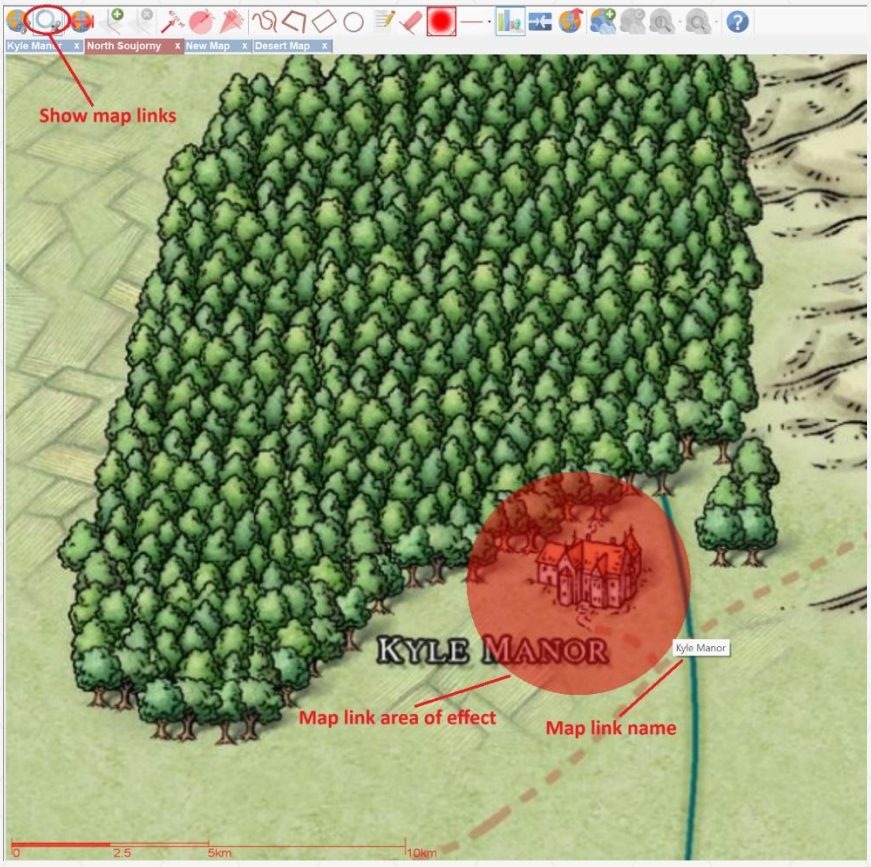
Map links can be deleted by right clicking on them and selecting the **Delete Map Link** option:



This will remove the map link from the map.

Making map links visible

By default, a map link's area is not displayed on the map, this is to allow a map to retain its aesthetic stylings. The user is only aware of a map link's presence by the tool-tip appearing. However, Sojour has a **Show Map Links** button which can be toggled on and off. This is shown activated below:



When toggled on, all map links on the map will have their area of effect shown as a shaded red circle. Moving the mouse cursor over this shaded circle will show a tool tip with the name of the map that's being linked to.

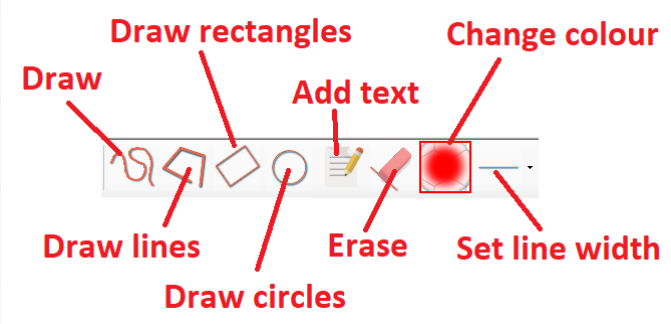
Double clicking in this area will open the linked map.

Map drawing tools

Sojour provides a variety of map drawing tools to allow you to mark up your maps in various ways.

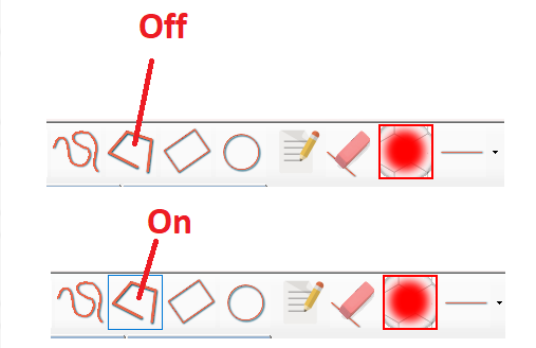
The tools are a little crude, but they do allow you create those quick-and-dirty maps for those random encounters that often happen in role playing games. Alternatively, they can also be used to annotate existing maps.

The tools that are available are as follows:



All map drawing tools work by being toggled on or off. This is achieved by clicking the relevant button on the map toolbar. Only one tool can be active at any one time.

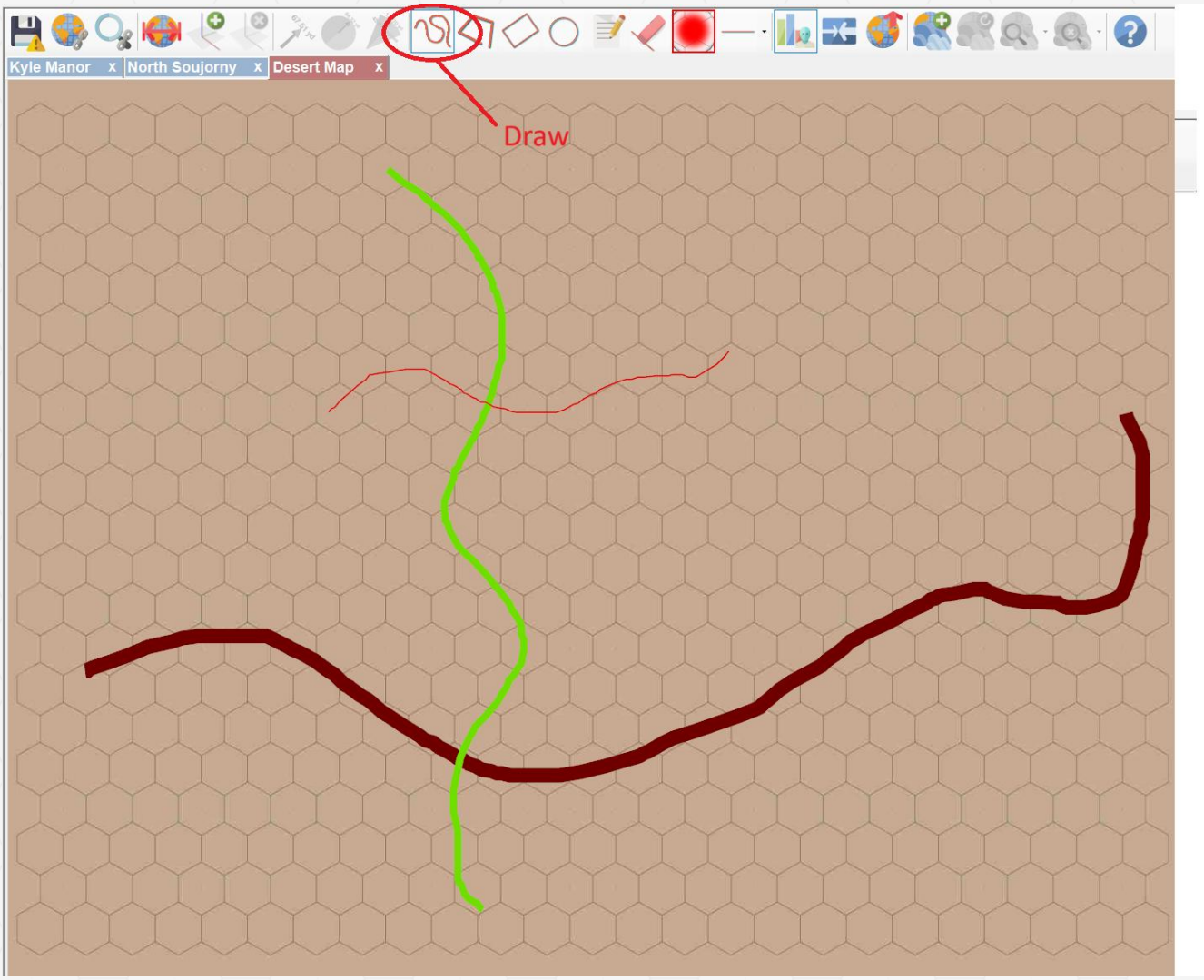
When a tool is toggled on, the tool's menu button will have a blue border around it:



In addition to the blue border, the mouse cursor will change from a crosshair to something more appropriate when hovered over the map window.

Pressing the <esc> key, or selecting another drawing or measurement mode will toggle off the current drawing mode.

Draw (aka Free Draw)



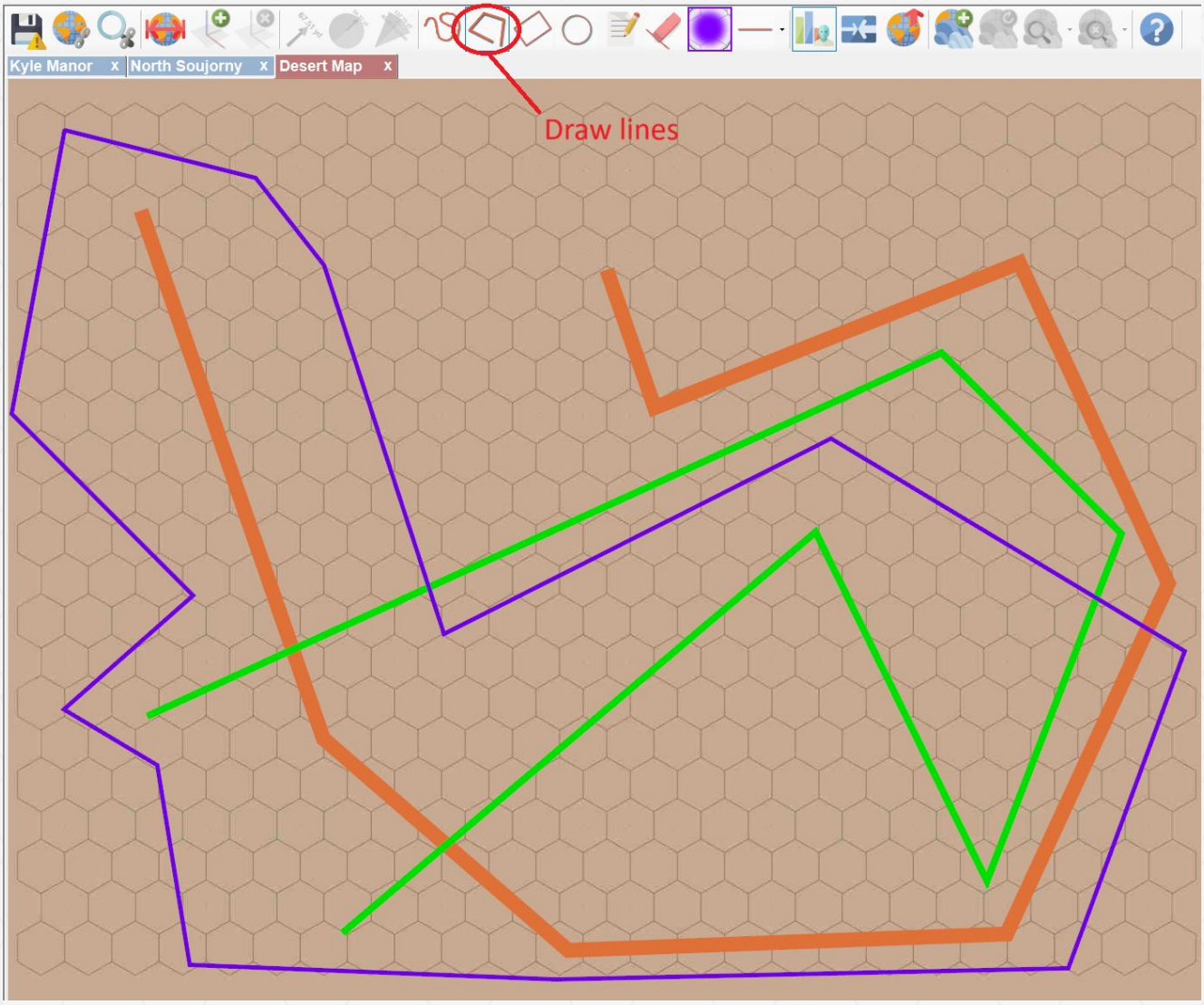
Draw is used to create freehand drawings. They can be made in any colour and in one of 5 line thicknesses as shown above.

To draw, click and hold down the left mouse button and then drag the mouse around to make the drawing.

Releasing the left mouse button ends the drawing of the current line. Once drawn, a line cannot be edited. It can only be erased.

Drawn lines become part of the map and will pan and zoom with the rest of the map.

Draw lines



Draw lines is used when there is a need to create drawings composed of straight lines only. They can be made in any colour and in one of 5 line thicknesses as shown above.

To use the draw lines mode, select it then click and release the left mouse button at the point where you want the first line to start. A line of your chosen colour and width will now rubber band from your starting location to wherever the mouse is moved.

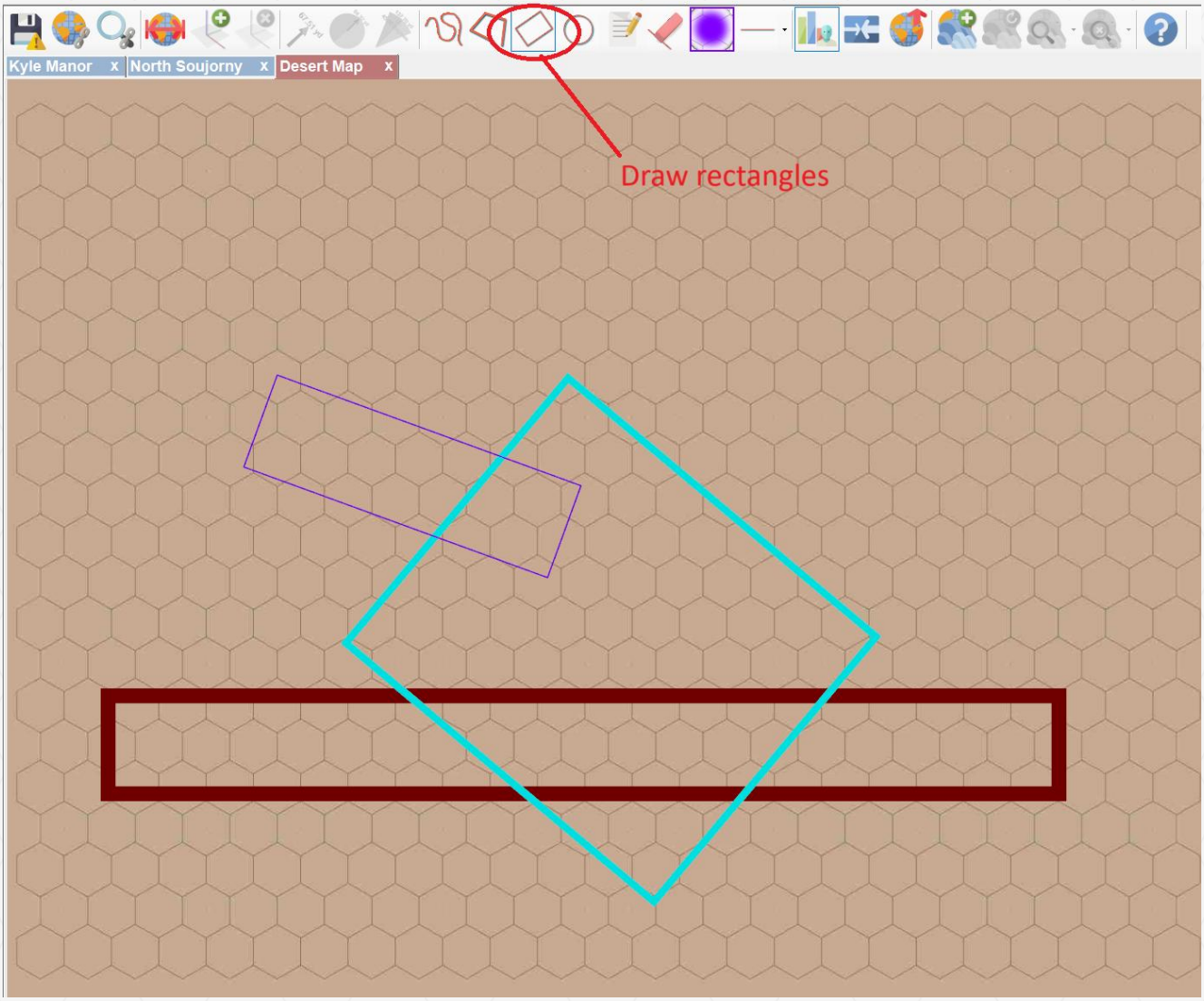
Left clicking and releasing the mouse again, will create another point on your line from where a new rubber banded line will originate.

Once you have drawn all the lines that are needed, right click the mouse. This will remove the rubber banding line attached to the mouse and will enable you to start drawing a brand new line.

Once a line is created it cannot be edited. It can only be erased.

Drawn lines become part of the map and will pan and zoom with the rest of the map.

Draw rectangles



Draw rectangles is used to draw rectangles or squares of any colour using one of five different widths. They can be drawn in any orientation and size.

To draw a rectangle, select the tool and the left click and release the mouse button. This will produce a rectangle using the colour and line width that you have selected. This rectangle is anchored to your mouse and not the map.

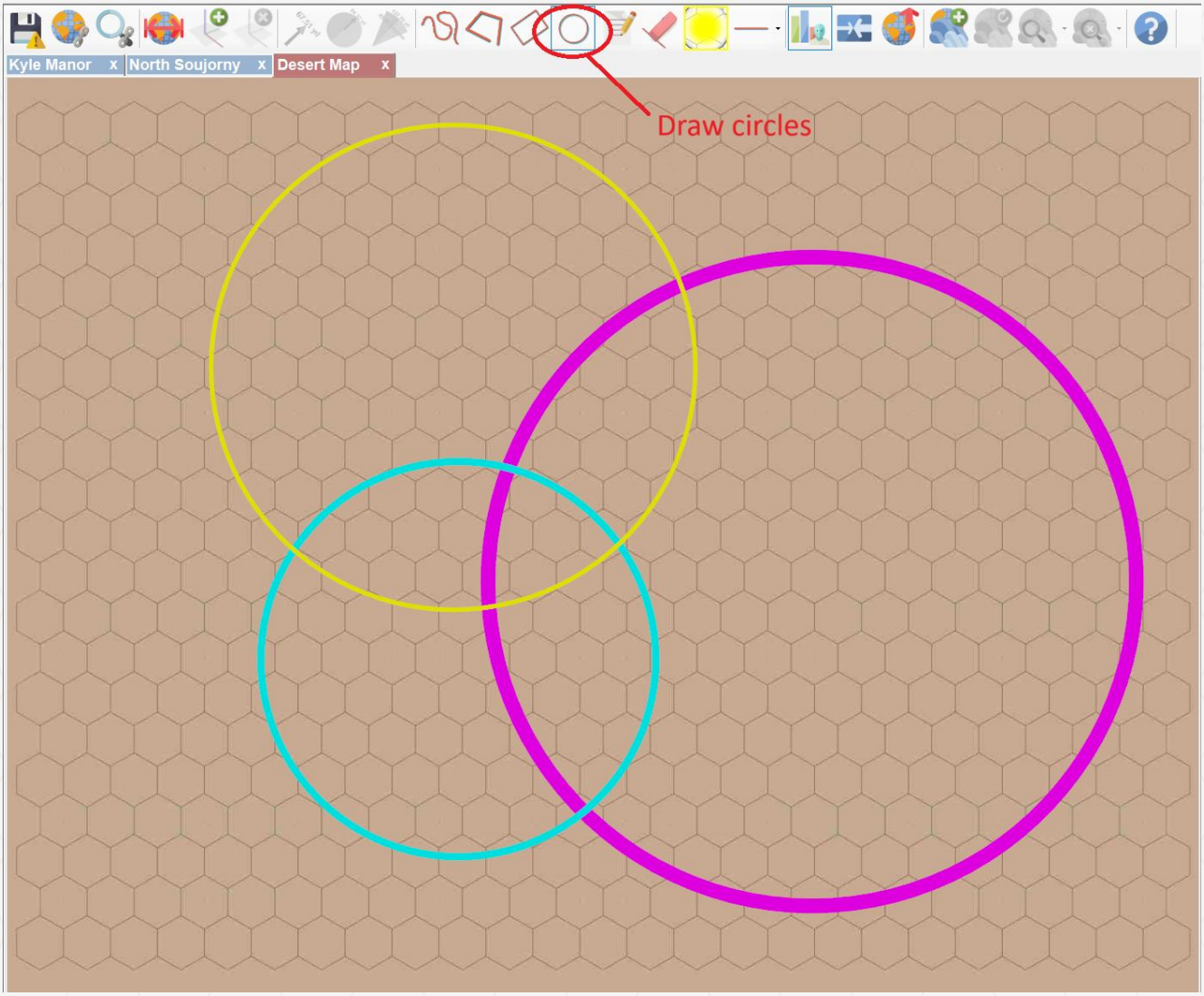
The mouse wheel is used to rotate the rectangle. <Ctrl> mouse wheel is used to increase the width of the rectangle, whilst <Shift> mouse wheel is used to change its height.

Once the rectangle is in the size and orientation that you want, move it to the location of the map where you want to place it, then click and release the mouse left mouse button. This will anchor the rectangle to the map.

Once drawn, rectangles are not editable, though they can be erased.

Drawn rectangles become part of the map and will pan and zoom with the rest of the map.

Draw circles



Draw circles is used to draw circles of any colour using one of five different widths.

To draw a circle, select the tool and the left click and release the mouse button. This will produce a circle using the colour and line width that you have selected. This circle is anchored to your mouse and not the map.

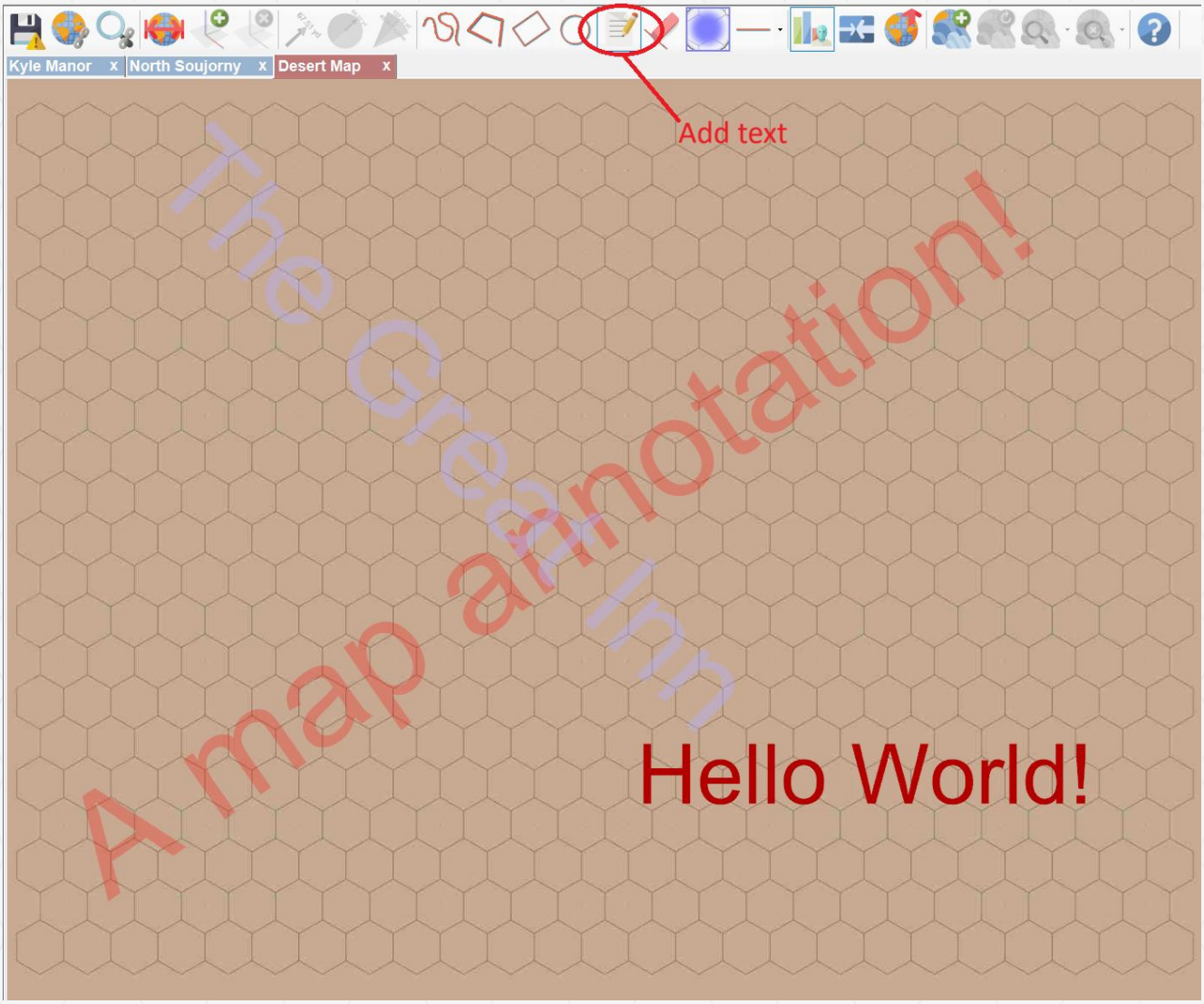
The mouse wheel is used to change the circle's radius.

Once the circle is in the size and position that you want, click and release the mouse left mouse button to anchor it to the map.

Once drawn, circles are not editable, they can only be erased.

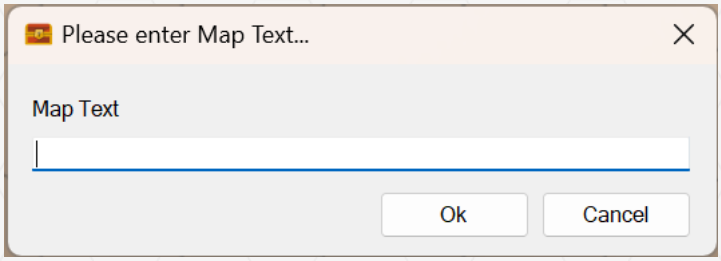
Drawn circles become part of the map and will pan and zoom with the rest of the map.

Add Text



Add text is used to annotate your maps with text. Text can be of any colour and size. It can also be made semi-transparent. The only real limitation is that it can only be a single line of text.

To add text, select the tool then click anywhere on the map. This will result in the window below being displayed:



Type in the text you want to annotate the map with, then click the Ok button. This will result in your text being attached to the mouse in the currently selected colour.

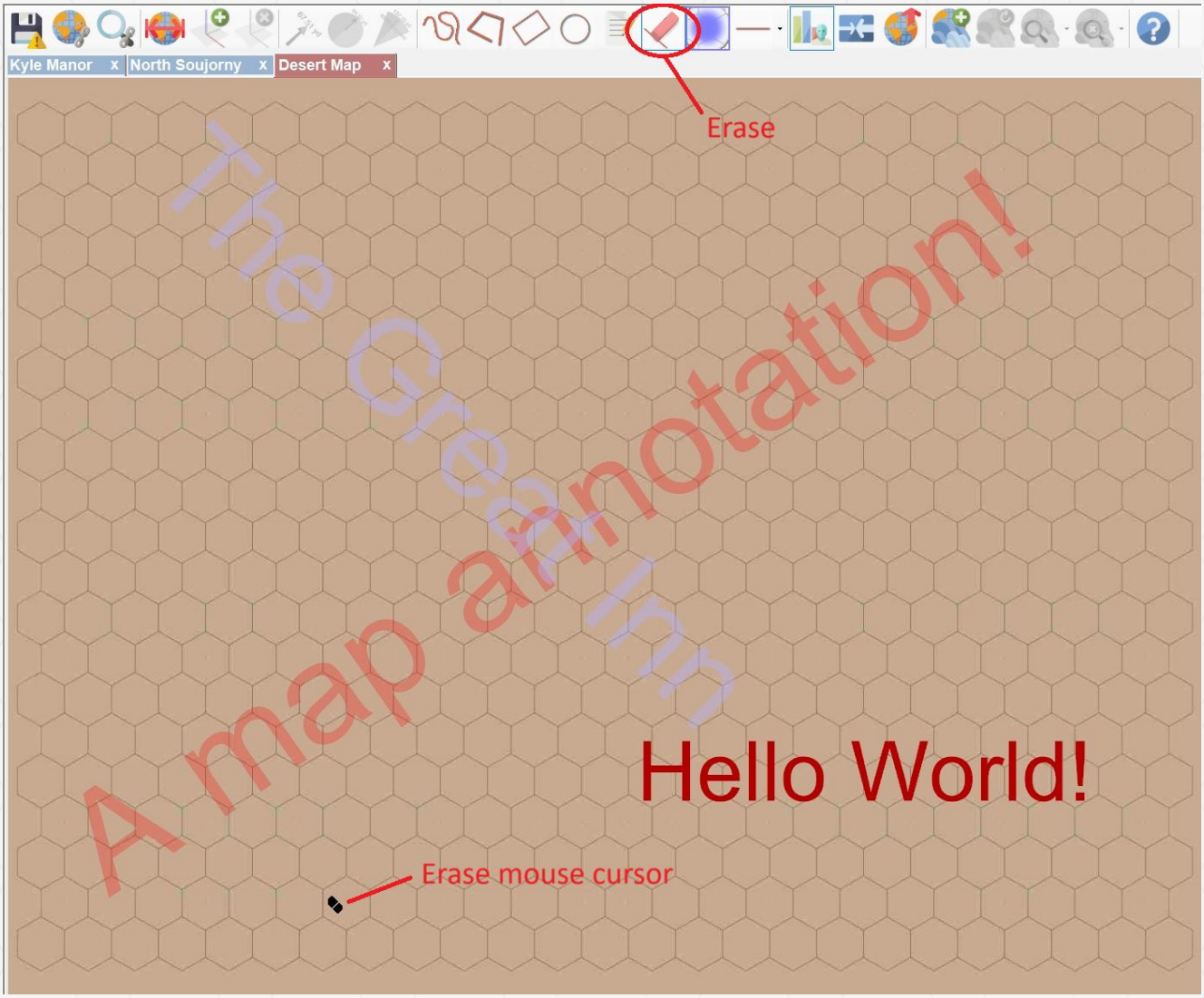
The mouse wheel is used to resize the text. <Ctrl> mouse wheel is used to rotate the text and <Shift> mouse wheel is used to change the text's opacity.

Once the text is in the size, position and opacity that you want, click and release the left mouse button to anchor it to the map.

Once drawn, text is not editable, though it can be erased.

Text becomes part of the map and will pan and zoom with the rest of the map.

Erase

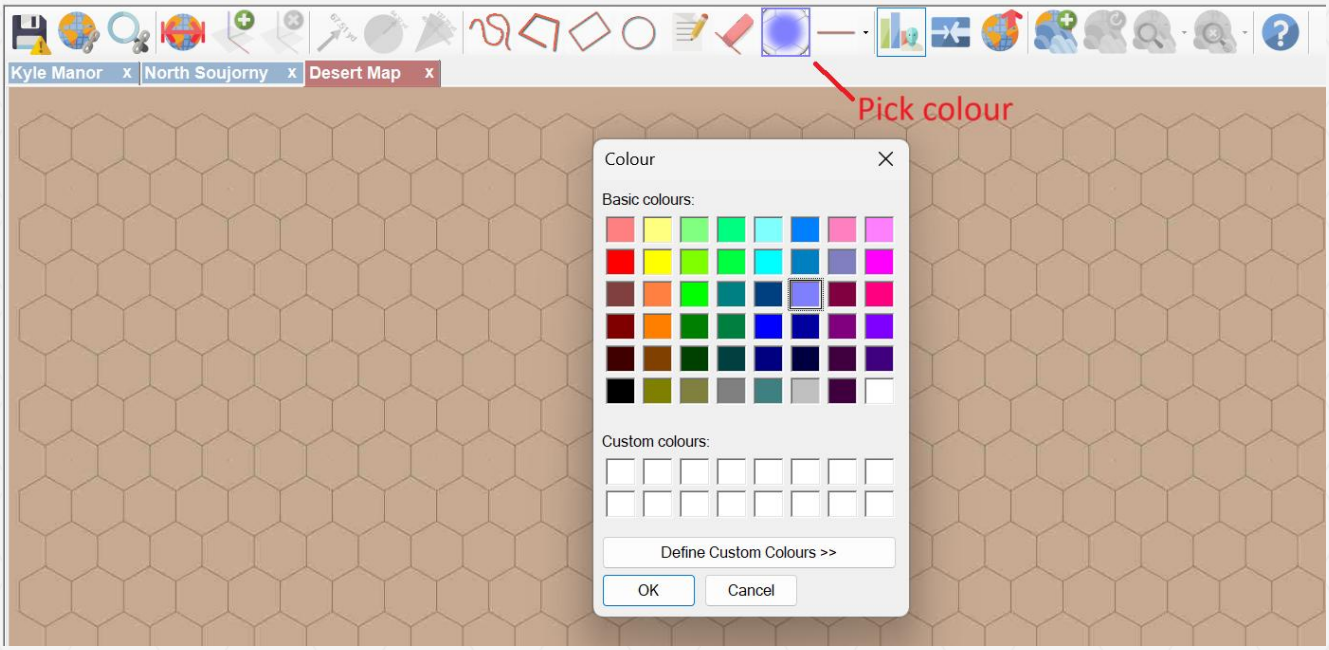


The erase button is used to erase any kind of drawing from the map. This includes text, circles, rectangles, drawn lines and freehand drawings.

When the erase mode is activated the cursor will change into the shape of an eraser as shown in the screenshot above.

To erase a drawing, left click on it with the eraser and this will remove it from the map.

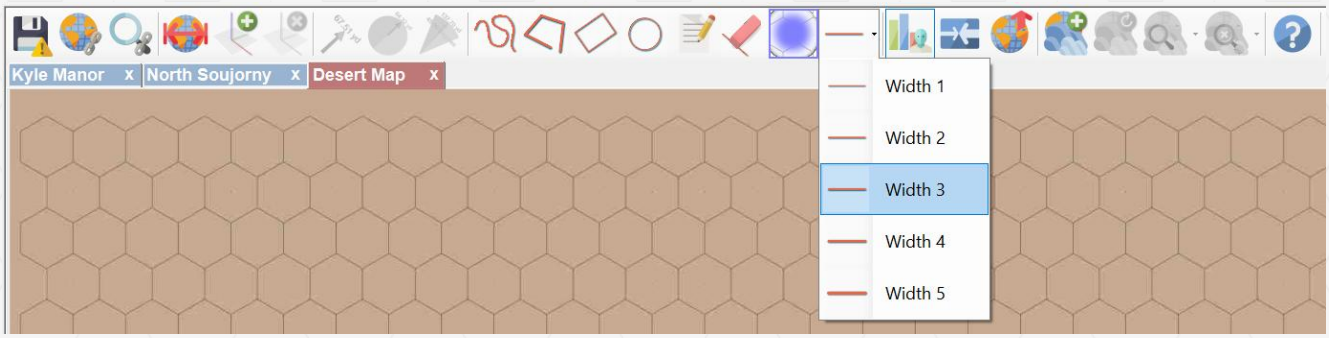
Pick Colour



The pick colour button is used to pick the colour of the next thing to be drawn. The colour of the button represents the colour that will be used.

Clicking the button will display the standard Windows colour selection dialog window as shown above. Simply pick the colour you want to use then click Ok.

Set Line Width



The **Set Line Width** button lets you select the width of the line, rectangle or circle that you are going to draw next.

The downward arrow to the right of the button opens up the line width drop down list as shown above.

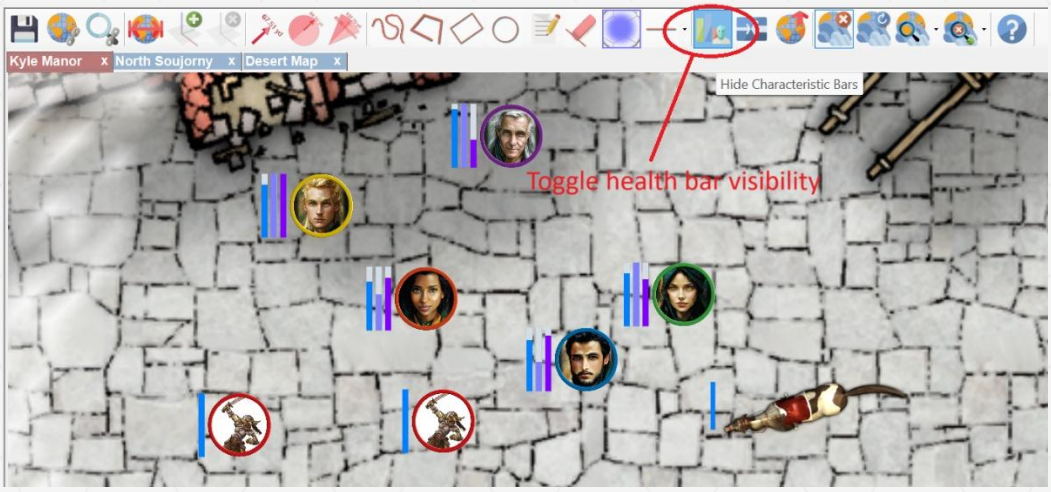
Sojour provides five different line widths and defaults to **Width 1** which is one pixel in width.

The five widths are defined as follows in Sojour:

Line Width	Pixel Width
Width 1	1
Width 2	3
Width 3	5
Width 4	8
Width 5	10

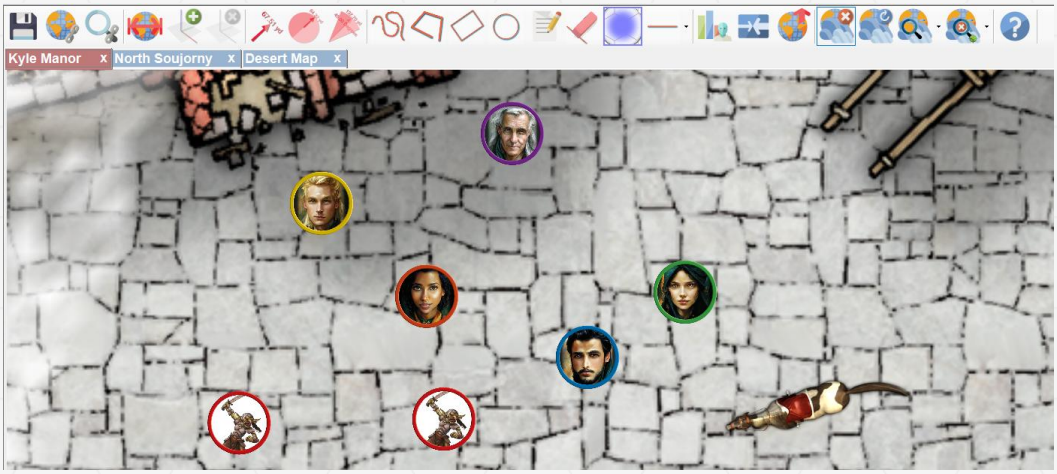
Setting the width will affect the next drawn freehand line, line, rectangle or circle.

Toggling map token health bars



Health bars for all tokens on a map can be toggled on and off by clicking the toggle button. Each map remembers this setting and persists it.

Here is the same map with the tokens toggled off:

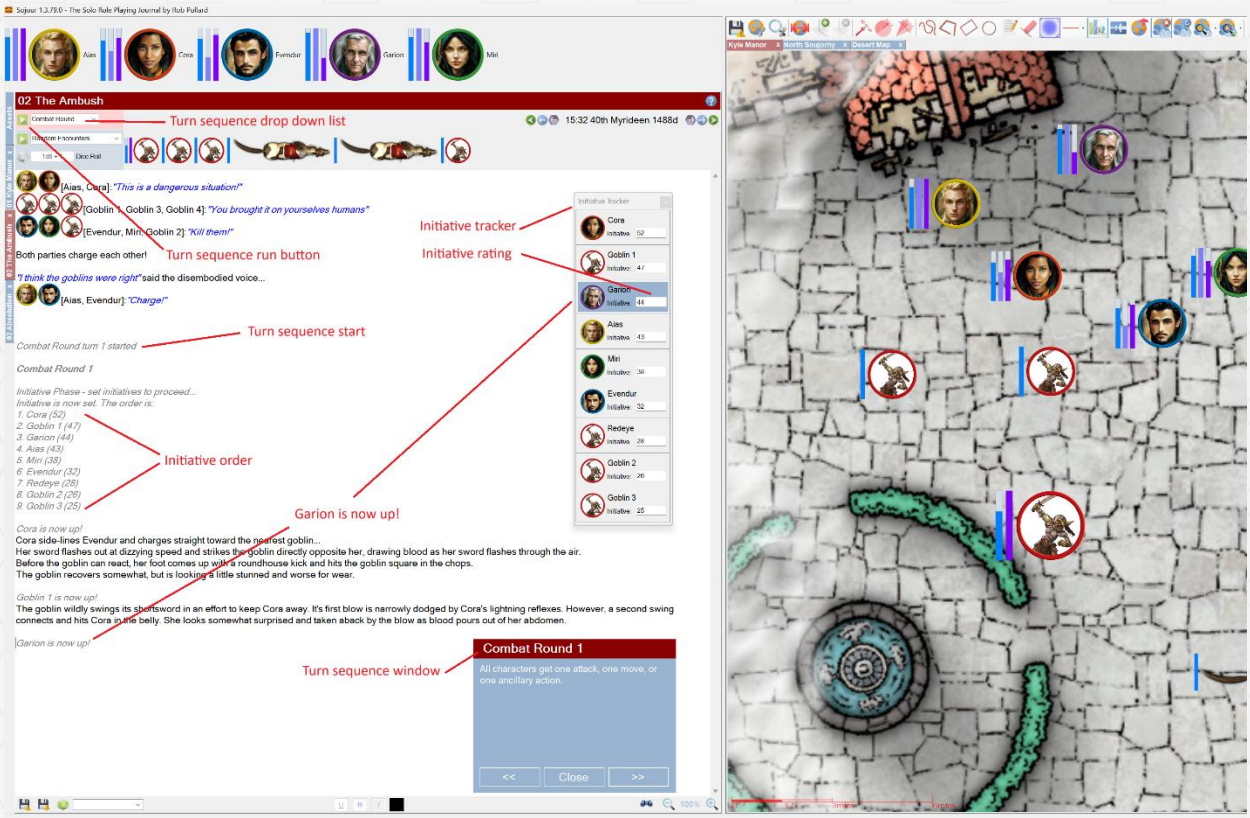


Turn sequences

Overview

Sojour provides a concept of turn sequences that allows the user to model turns. The user can then run these turns to keep track of that turn and the elapsed time within the journal. Turn sequences can also have an optional initiative tracker enabled for them.

A picture paints a thousand words, so here is a screenshot of a turn sequence in action:



In the above screenshot a turn sequence has been created to model a Windows Guardians combat round complete with initiative tracker. It has been run for an encounter. The grey italicised text in the journal was created by the turn sequence, the rest of the text was created by the player as they role played the encounter.

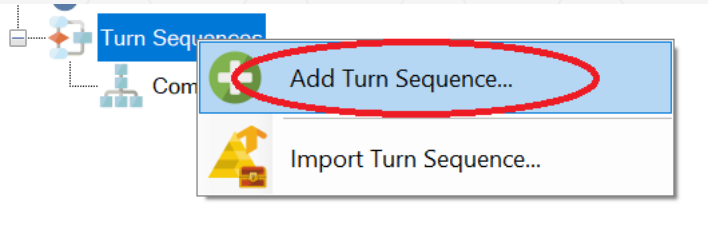
A running turn sequence consists of up to four parts:

1. **Turn sequence window:** This window shows the current turn and optionally phase. It generally contains information to act as an aide memoire for the user with regard to the running of the turn or phase.
2. **Journal text updates:** The journal will have many automatic updates in it created by the running turn sequence. The generated text is in light grey italicised text, which can be edited by the user if required. In the above screenshot we can see where the user has typed in their own text in black between the turn sequence entries as they play out the encounter.
3. **Calendar updates:** As each new turn is started, the campaign calendar will have its time updated to reflect how much time has elapsed so far. This elapsed time will be displayed in the journal.
4. **Optional initiative tracker:** If this is enabled for the turn sequence, it will appear and will automatically regulate the flow of combat once the initiatives are entered by the player.

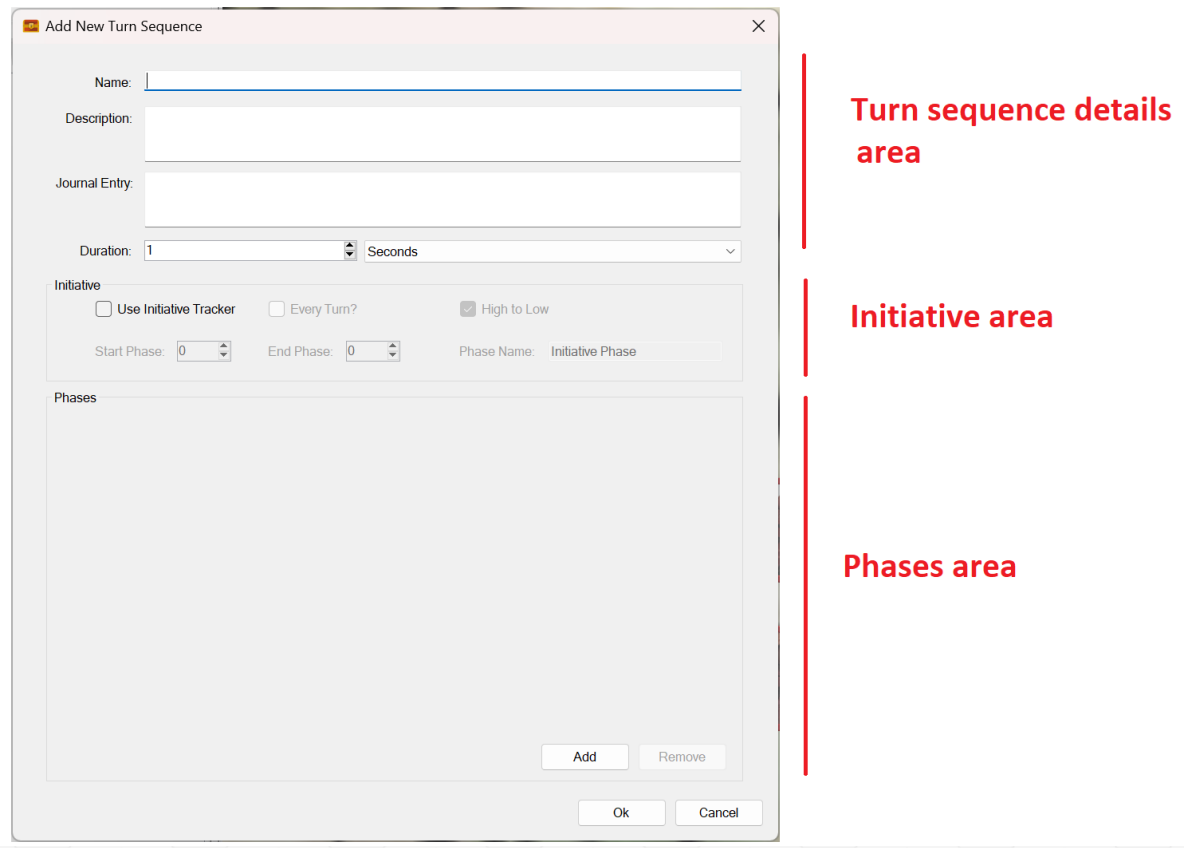
The turn sequence window and initiative tracker can be dragged anywhere on the desktop. Sojour will remember where these windows were left prior to closure. When a new turn sequence is run, those windows will appear in their last used locations.

Turn sequence creation

To create a new turn sequence, right click on a **Turn Sequences** node in the **Assets Browser** and select **Add Turn Sequence…**:



Once clicked, the **Add New Turn Sequence** dialog appears:



The window is broadly split into three horizontal areas from top to bottom:

1. **Turn sequence details area**: All of the fields related directly to the turn sequence are displayed here.
2. **Initiative area**: This boxed off area contains all the fields required for setting up the initiative tracker. Adding initiative to a turn sequence is optional and not required.
3. **Phases area**: This area allows the user to add one or more phases to their turn sequence to allow the modelling of complex turns. They are optional, in that the user can define a turn with no phases.

Turn sequence fields

1. **Name**: The name of the turn sequence. This will also show up in the journal when the turn sequence is stopped or started. It is also displayed as the title of the turn sequence window.
2. **Description**: This field appears in the turn sequence window. Use it to provide a description and aide memoire for the turn. The field is limited to 300 characters.
3. **Journal Entry**: This field is injected into the journal whenever a new turn is started. Normally left blank, but provided for those that want to add additional text to their journals. This field does not have an upper character limit as its text appears in the journal only which is not limited by space.
4. **Duration**: This is a two part field consisting of a time division (seconds, minutes, hours or days) and a numeric representing how much time. Use this field to define how long one turn lasts.

Initiative fields

Keeping track of initiative during a turn is optional. All the initiative fields will be disabled until the **Use Initiative Tracker** option is selected.

1. **Use Initiative Tracker**: This field enables or disables the use of the initiative tracker. The other initiative fields will be enabled or disabled dependent on this setting.
2. **Every Turn?** This field determines whether initiative needs to be set once at the beginning of the first turn only, or once for every subsequent turn too.
3. **High to Low**: This field determines how the initiatives are to be ordered. It defaults from High to Low where higher initiatives go before lower initiatives. This can be reversed by unticking the checkbox.
4. **Phase Name**: The name to be used for the initiative phase. This name will show up in the journal.
5. **Start Phase & End Phase**: These two fields are only used when a turn sequence has phases defined for it, otherwise it is disabled. These two fields determine the contiguous phases where initiative will apply. See the Initiative with Phases section for a more detailed explanation.

Phases Area

This section allows the user to add one or more phases to the modelling of their turn.

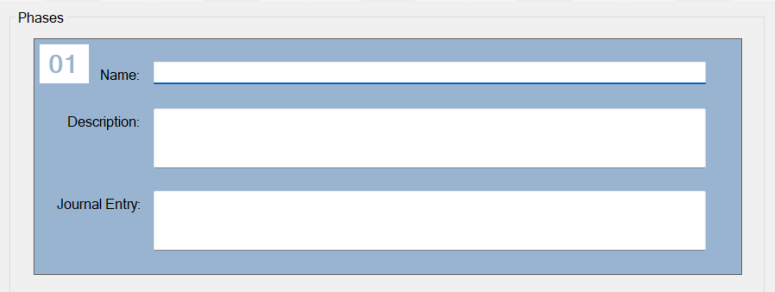
Each phase has an identical set of fields:

The phase fields are similar to the parent turn sequence fields:

1. **Phase Number**: This number represents where in the order of phases that this phase will be run. Smaller numbers are run first. It cannot be changed. If initiative is active for this phase, the number will have a red background, otherwise it will have the light blue background shown above. The initiative **Start Phase** and **End Phase** fields are used to adjust this.
2. **Name**: The name of the phase. This name will show up in the journal and turn sequence window whenever that phase is run.

3. **Description**: This field will appear in the turn sequence window when this phase is active. Use it to provide a description and aide memoire for the phase. The field is limited to 300 characters.
4. **Journal Entry**: This field is injected into the journal whenever this phase is active. Normally left blank, but provided for those that want to add more text to their journals. This field does not have an upper character limit as its text only appears in the journal, which is not space limited.

A phase can be deleted by selecting it and using the **Remove** button. A phase can be selected by clicking on any non-field part of it. When selected, its background colour will change to light blue:



The phase can be deselected by clicking it again.

Initiative with Phases

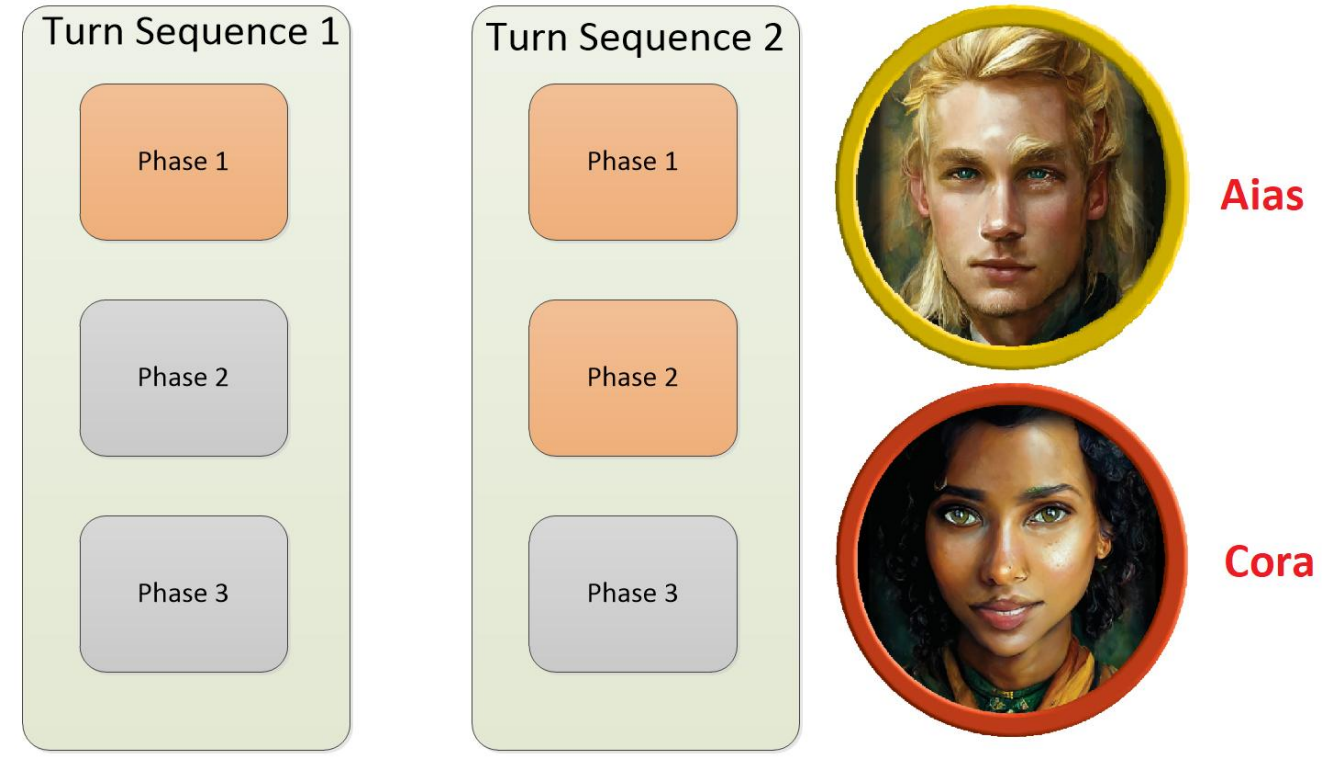
Where a turn sequence has initiative enabled and no phases, Sojour will cycle through each character and npc one at a time in their initiative order. Once they have all been, the next turn starts.

However, once phases are added, the user has to decide which phases are bound to initiative and which are not. Phases are bound to initiative in a single block of phases determined by the **Start Phase** and **End Phase** fields. An initiative block of phases can consist of one or more phases.

When a turn starts, the characters will be cycled through at the start of the initiative block. Sojour will then loop through the initiative block's phases with the *same* character. Once all the phases in the initiative block have been run, Sojour will go back to the first phase of the initiative block and then pick the next character, unless all characters have been. In that case Sojour will move on to any non-initiative phases that follow.

All non-initiative phases will just run and will have no characters selected or associated with them.

The above can be hard to grasp without seeing it in action, so here are two concrete examples:



In the above diagram we have two turn sequences that use initiative, each has three phases. We also have two characters on the map, one called Aias and the other called Cora.

A turn that has phases and no initiative (not shown above) would run as written: Phase 1, Phase 2, Phase 3 and then on to the next turn, taking you back to Phase 1.

However, once initiative is enabled, the user will get many options dependent on how they set the **Start Phase** and **End Phase** settings for the initiative.

In the diagram above**, Turn Sequence 1** has both the **Start Phase** and **End Phase** set to 1. **Turn Sequence 2**, on the right, has its **Start Phase** set to 1 and its **End Phase** set to 2.

When **Turn Sequence 1** is activated, it will run as follows:

1. Aias Phase 1
2. Cora Phase 1
3. Phase 2
4. Phase 3
5. Next turn, back to phase 1 (step 1 above)

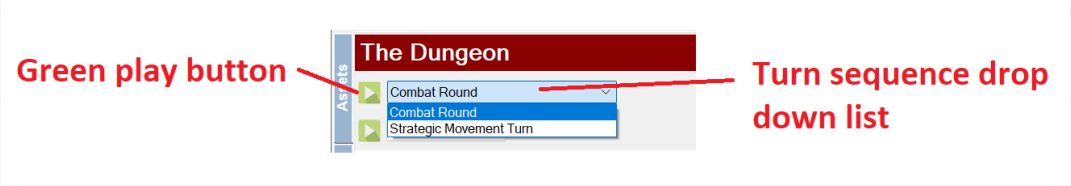
Running **Turn Sequence 2** will result in:

1. Aias Phase 1
2. Aias Phase 2
3. Cora Phase 1
4. Cora Phase 2
5. Phase 3
6. Next turn, back to phase 1 (step 1 above)

The user is free to bind any phases to initiative using the **Start Phase** and **End Phase** settings under the initiative section. The only limitation is that the selected phases must be a contiguous block and that there can only be one block.

Running a turn sequence

Turn sequences are run from any open journal. To run one, select which turn sequence you want from the drop down list, then click the green play button alongside it:



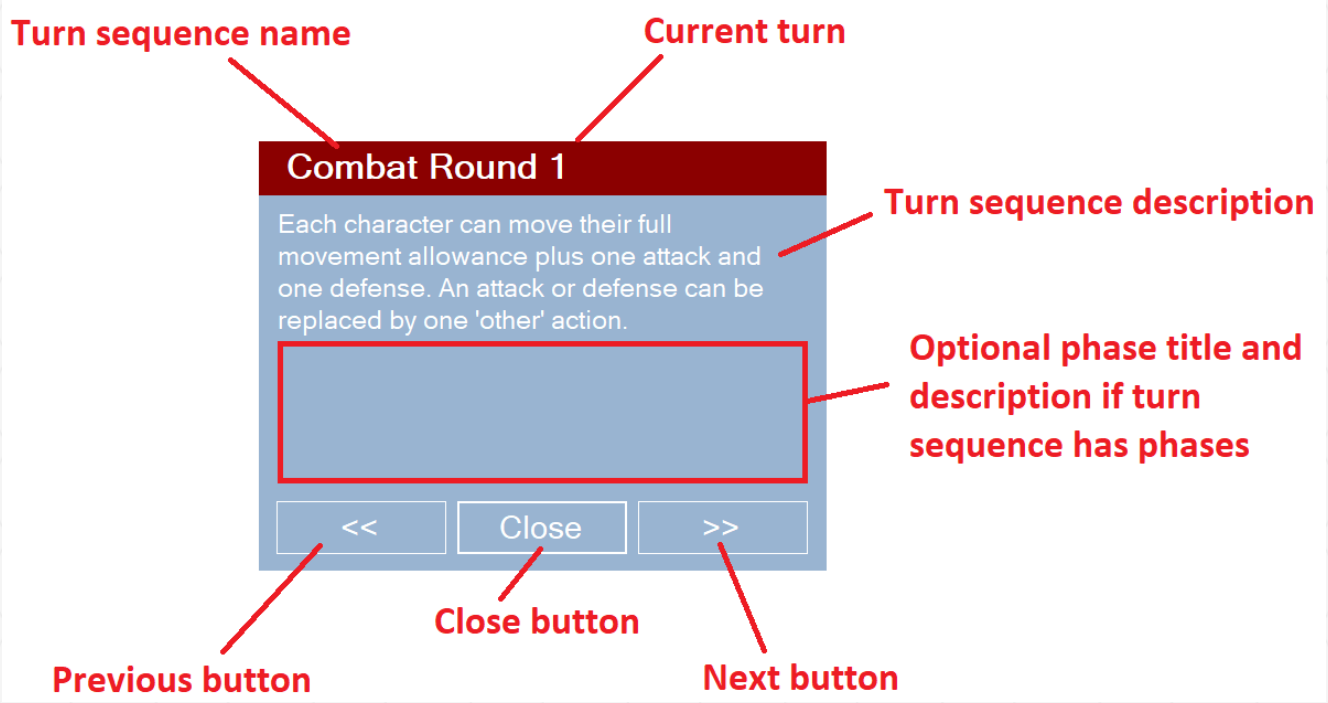
Once run, the turn sequence window will be displayed. If initiative tracking has been enabled and there are one or more characters and/or npc's on the map, an initiative tracking window will also be displayed:

Both windows can be dragged to wherever the user prefers. Both windows will remember their last used position and will reappear there next time a turn sequence is used.

If the activated turn sequence uses initiative, the user will be prompted to enter the initiative values into the initiative tracker before proceeding. This can be seen in the screenshot above.

Warning! Turn sequences that have initiative won't allow one to proceed to the next turn unless there is an open map with at least one or more characters and/or NPCs on it. In this situation it can only be closed.

Turn sequence window



The turn sequence window displays the current turn and optional phase along with some aide memoire text. It is this window that controls the flow of the turn. Turns can be moved to the next or previous phase, or alternatively, they can be closed.

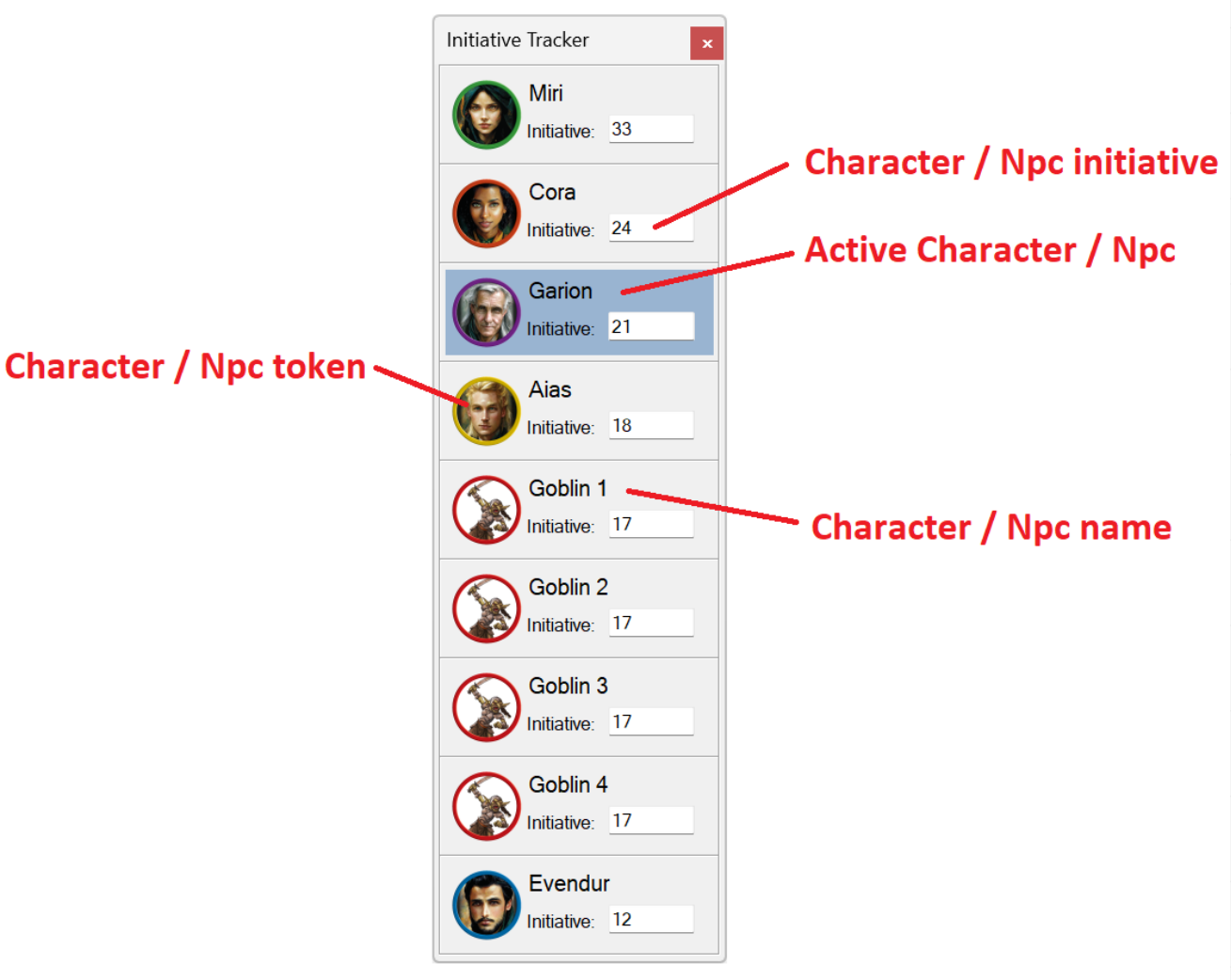
The aide memoire text comes from the turn sequence description.

The previous and next buttons are used to move the turn forwards or backwards. These buttons will be disabled if the turn sequence is using initiative but the initiatives have not yet been set.

The close button is the only way to exit from a running turn sequence.

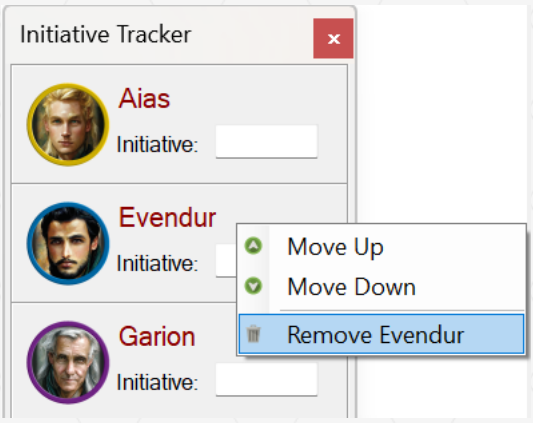
The turn sequence window can be dragged anywhere on the desktop by left clicking on it and dragging it to the required location. Sojour remembers this location. When a turn sequence is next run, the turn sequence window will appear in the last place it was left.

Initiative tracker



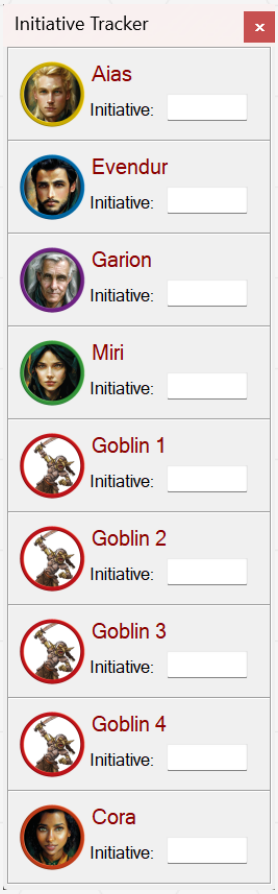
The initiative tracker is used to keep track of which character or npc is currently up for action. In the above screenshot we can see that Garion is up.

The initiative tracker will be populated by all npcs and characters that are on the current opened map. Dead npcs will not appear. Sojour allows the user to remove npcs and characters from the initiative tracker by right clicking them and selecting **Remove** <<character/npc name>>:



Once removed characters/npcs cannot be re-added.

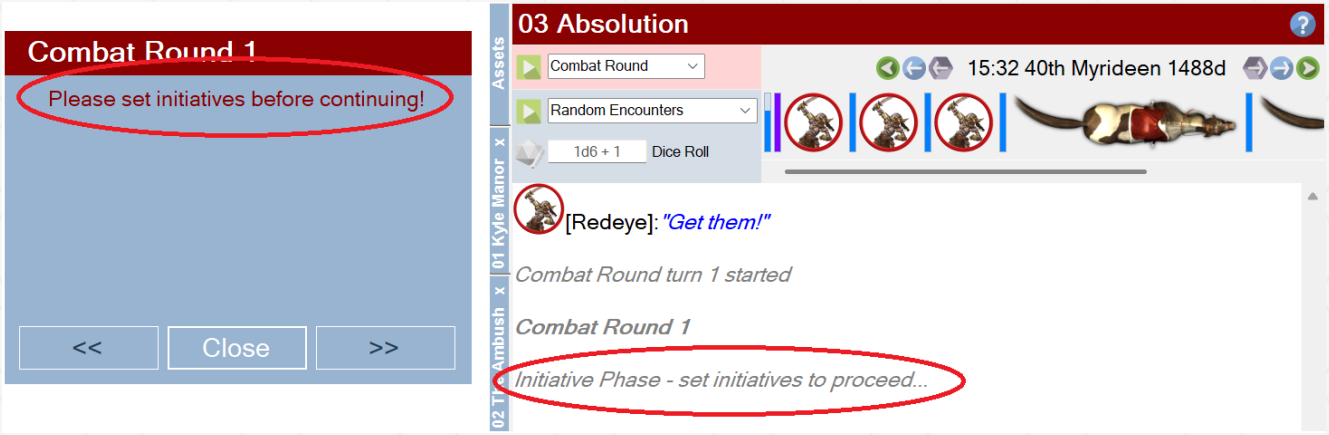
When the initiative tracker first appears, none of the initiatives will be set and it will look like this:



All initiative fields will be empty and the character/npc names will be coloured red to let you know their initiatives have not been set.

Warning! A turn cannot proceed until all of the initiative values have been entered.

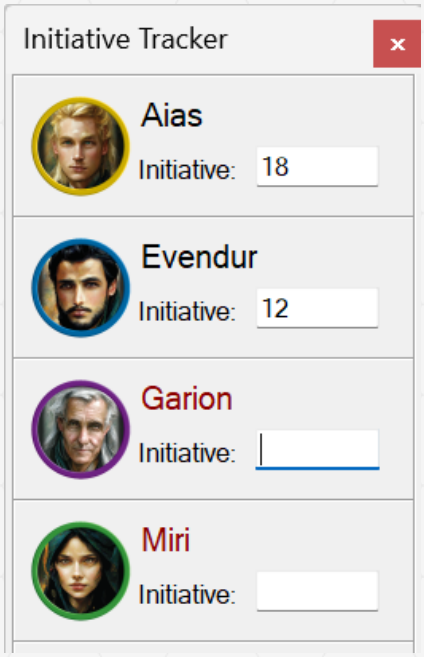
Sojour provides two hints that the initiatives need to be set:



Firstly, the turn sequence window will remind the user to set initiatives and secondly, the active journal will have text added to it to inform the user to set initiatives.

The turn sequence's next and previous buttons will be coloured black and will be disabled until all the initiative values are entered.

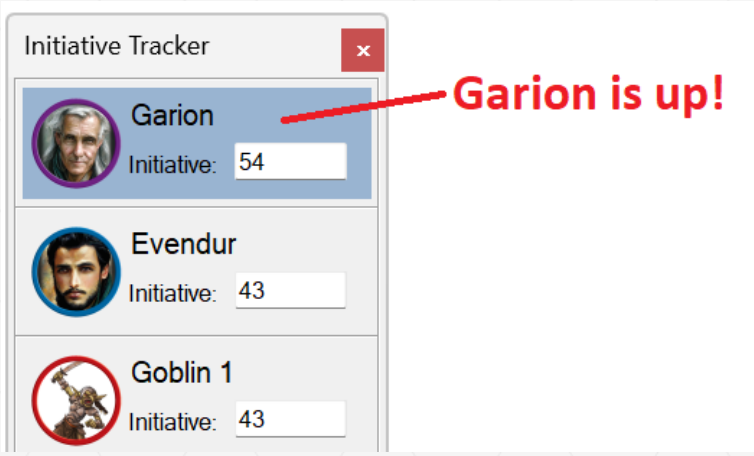
As the initiatives are entered, that character's or npc's name will change to black within the initiative tracker:



The **Tab** button can be used to move to the next character or npc that needs its initiative setting. Tabbing out of the initiative field will set that character's or npc's initiative. Alternatively, waiting a few seconds will also set the initiative.

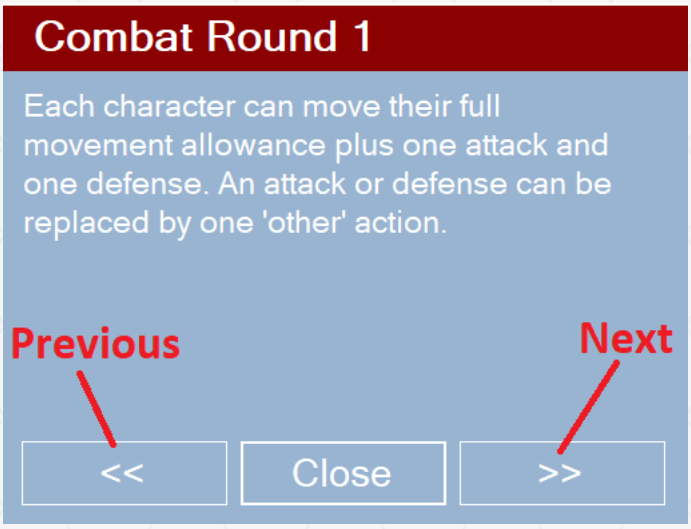
The initiative tracker will automatically order characters by their initiative values. By default it sorts the characters from high to low initiatives. The order can be changed so as to sort the characters from low to high initiatives by unticking the turn sequence's **High to Low** check box.

Once all the initiatives are set, a character/npc will be highlighted in light blue to show that it is the one whose turn it is to move:

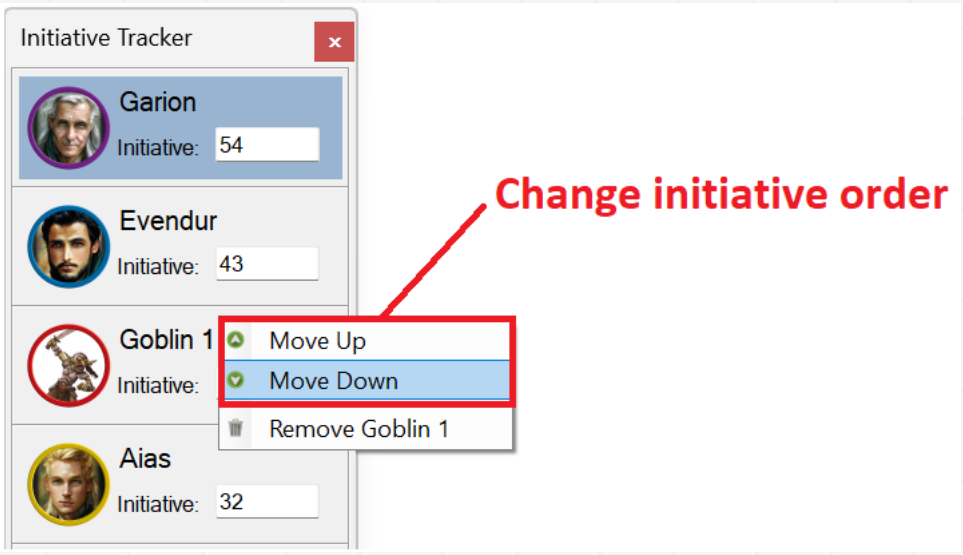


The map will also briefly draw a red ring around its token to highlight it. If **Auto Centre on Tokens** is enabled under settings, the map will also centre on the character/npc.

The current character/npc's initiative can be moved to the next or previous character/npc using the next and previous buttons on the turn sequence window:



Characters and npcs can have their initiative order changed either by typing a new initiative value or by right clicking on the character/npc and selecting either **Move Up** or **Move Down**:

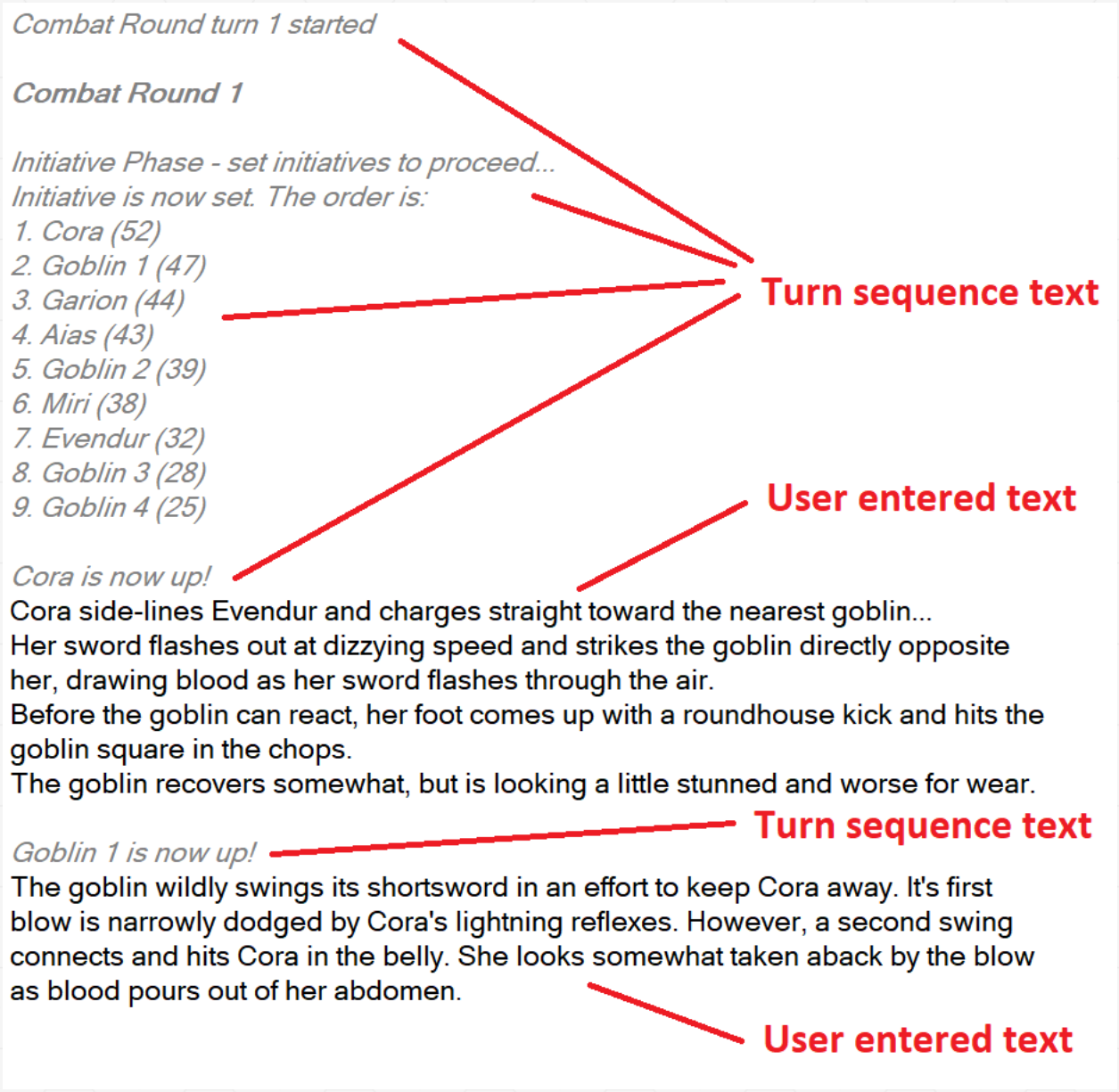


Npcs will be automatically hidden from the tracker when their hitpoints dip to zero or below. If their hitpoints are recovered, the npc will reappear on the initiative tracker in the same place they were when they died.

Journal updates

A running turn sequence will update the journal as the turn is run. This information will cover which turn is currently running, the initiative order, which characters or npcs are currently up and also how much time has elapsed.

All entries injected into the journal are written using light grey italics to differentiate it from user input:



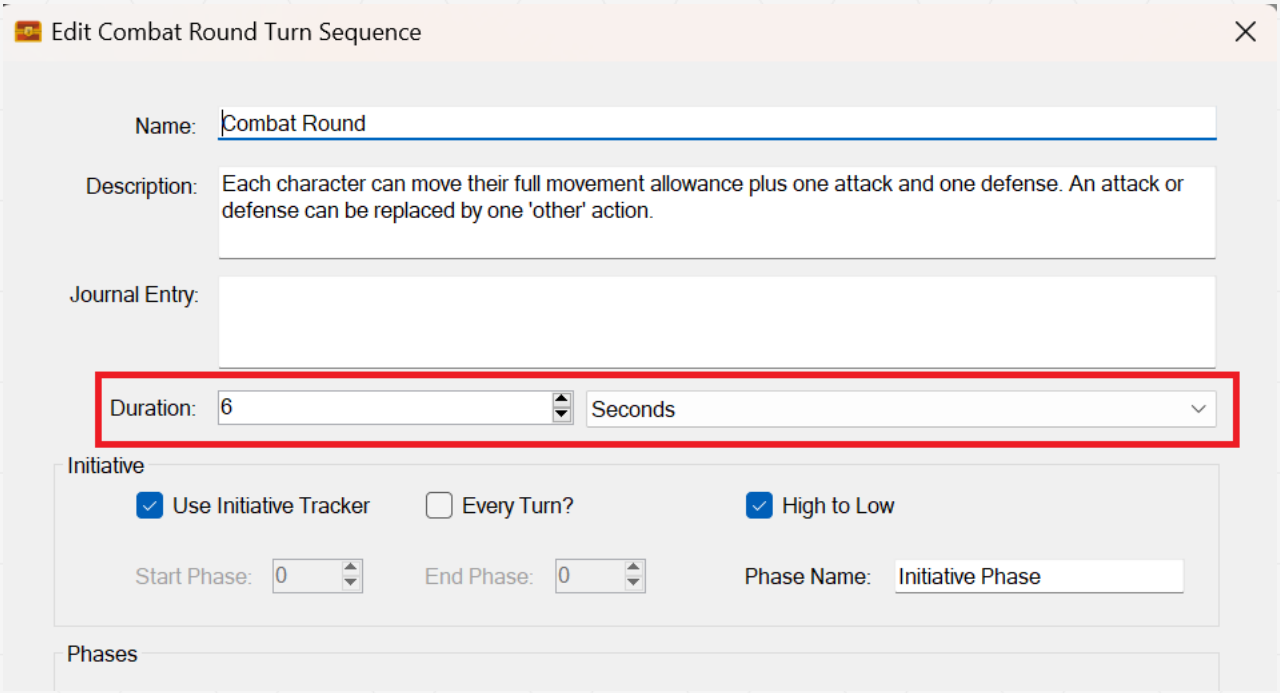
The injected text from the turn sequence is fully editable by the user.

Calendar updates

Turn sequences will automatically update your campaign's time as each turn is progressed. You can see the changes to the time by observing the entries made in the journal by the running turn sequence:

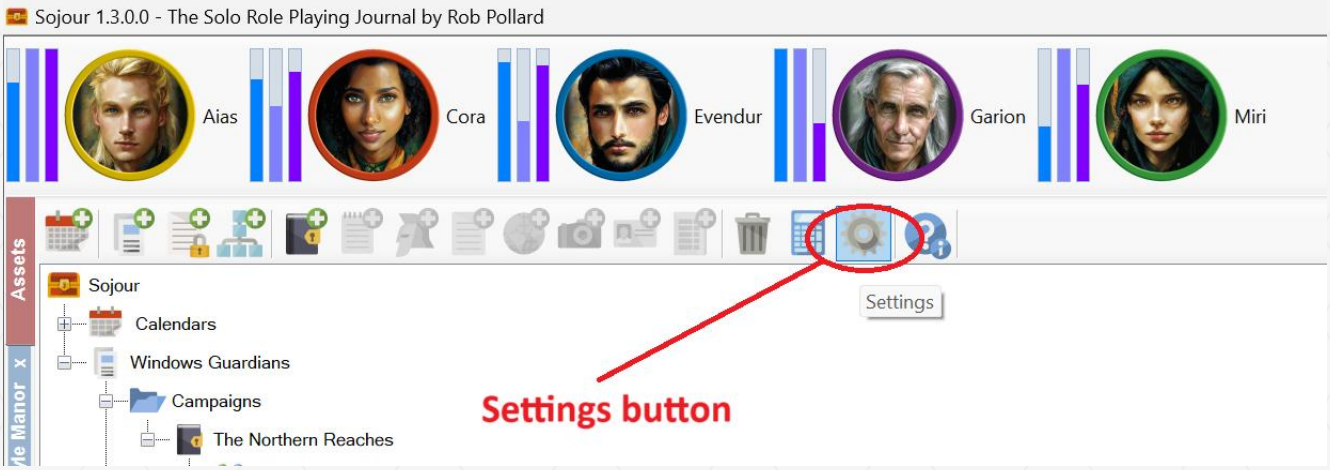
In the above example 6 seconds have gone by as the first combat round has ended and the second one starts. This change in time will also be reflected in the campaign calendar.

The amount by which the calendar's time is altered is set by the **Duration** settings in the turn sequence:

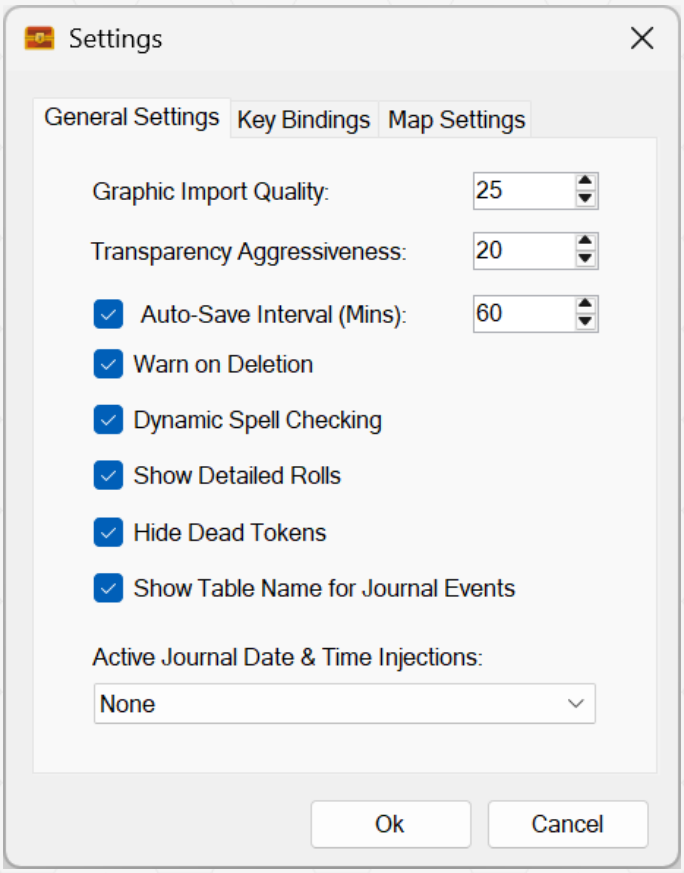


Settings

Sojour has a number of global settings that can be altered to the user's preferences. These settings are persistent and will be saved between Sojour instances. To get at these settings click the **Settings** button on the main toolbar:



Clicking the **Settings** button will result in the following window being opened:



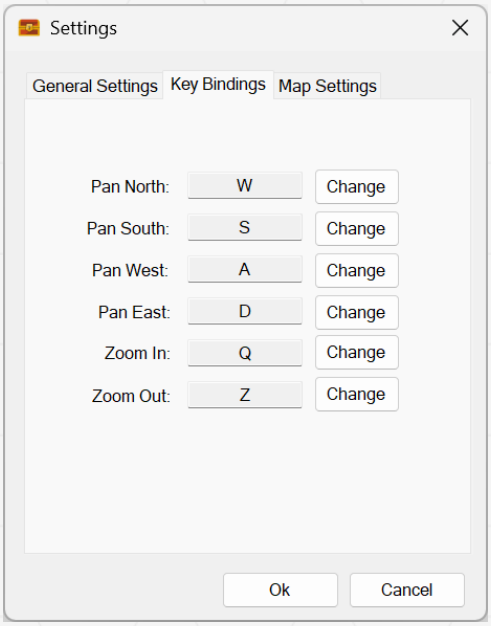
This window has three tabs and we will cover each in turn.

General Settings

The settings from top to bottom are:

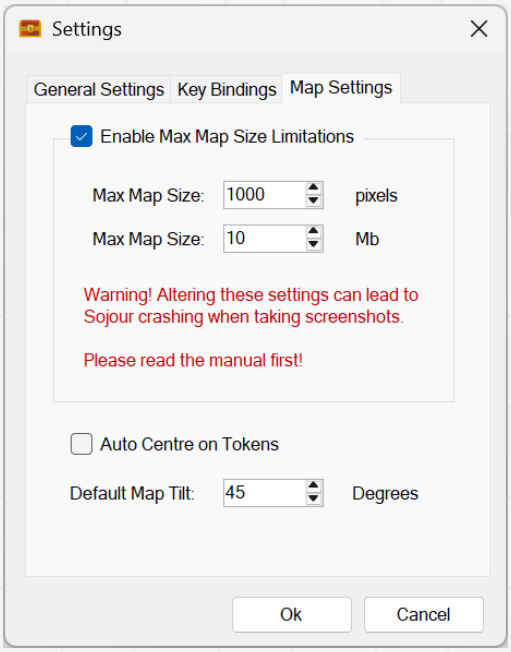
1. **Graphic Import Quality:** This setting determines how much compression is applied to images imported into Sojour. The numbers represent the final file size after compression. For example, the window is showing 25 which means that an imported image will be compressed to 25% of its size. 25% is Sojour's default value. Changing this setting trades image quality for memory usage.
2. **Transparency Aggressiveness**: This setting is used by the character/token transparency tool. This tool allows the user to pick a colour on a token to set as transparent. Sojour will pick that colour and other related nearby colours based on this aggressiveness setting. The higher the value, the greater the colour range that will be included as part of the transparency. The default value is 20. More information can be found here.
3. **Auto-Save Interval (mins)**: This setting determines how often Sojour will save your work in the background. It defaults to being enabled and set to once every hour. This is a fall-back feature as Sojour will automatically save your work on exit.
4. **Warn on Deletion**: This setting will determine if a confirmation dialog window is displayed when deleting an item in the **Assets Browser**. It defaults to being enabled.
5. **Dynamic Spell Checking**: This setting determines whether Sojour will temporarily highlight misspelt words in red. Some users may find this distracting so this option can be used to turn it off. Note, that even when off, the user still has access to spell checking facilities by right clicking on words within the journal. This setting defaults to on. More information can be found here.
6. **Show Detailed Rolls**: This setting determines whether detailed dice rolls are shown or not. It defaults to off. More information can be found here.
7. **Hide Dead Tokens**: When enabled, NPC tokens will be hidden from the map and removed from the initiative tracker when their hit points reach zero or less. When their hitpoints change back to above zero they will reappear. Note this setting does not affect current tokens on the map or the initiative tracker.
8. **Show Table Name for Journal Events**: This setting determines whether triggered table events show the originating table's name as part of the text injected into the journal. This setting is defaulted to on.
9. **Active Journal Date & Time Injections**: This setting determines what gets injected into the active journal when the campaign calendar's date or time are changed. By default this setting is set to None. More information can be found here.

Key Bindings



These allow the user to change the default key bindings for map navigation. These settings will be important for users not using the standard QWERTY keyboard. To change a mapping, click the **Change** button then press the key you want to use for that mapping. The defaults are shown in the screenshot above.

Map Settings



The limitation settings are described in more detail within the map limitations section and the map tilt settings are described in the map tilt section.

Auto Centre on Tokens will auto centre the map on the active token automatically if ticked. The active token will be the one that the mouse is waved over in the main character bar, an npc in the journal toolbar or the selected character/npc of a turn sequence.

Software crashes

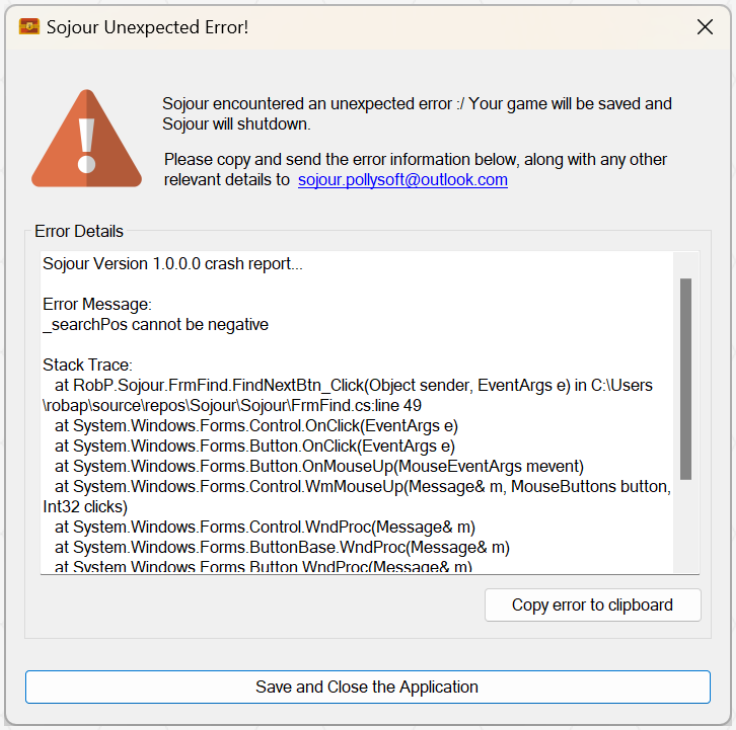
Despite my best efforts, Sojour may well eventually crash for you. Firstly, I apologize for your experience, but together we should be able to fix the underlying issue.

Crashes are a result of Sojour being a complex piece of software and there being a multitude of system configurations and ways to use Sojour. There is only one of me, and despite pouring over 200 hours into testing, there is a high likelihood that someone will find a way to break the system.

To that end, Sojour comes with built in crash protection.

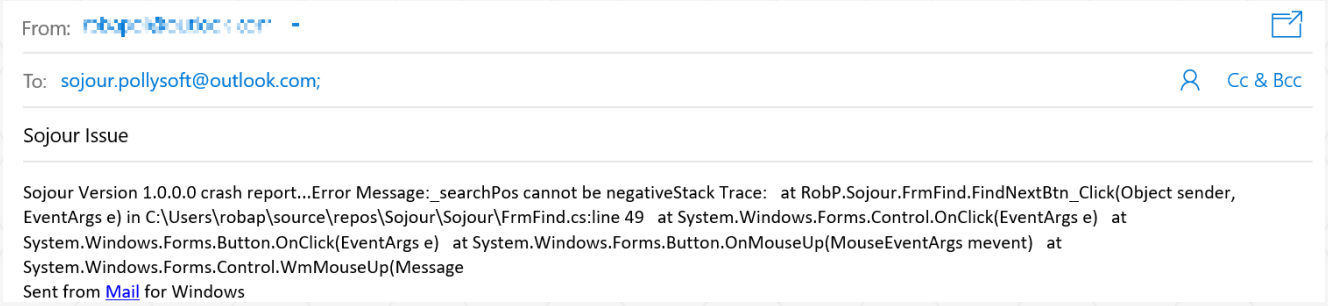
General Crashes

Should Sojour crash outside of loading, you will see this screen:



The screen will have a field containing the technical detail as to what went wrong, as well as a blue email link that you can click to email me.

Clicking that link will open your default email client and create an email that is ready to send. The email will automatically include the technical details:



No personal information is sent other than your email address.

You can help with the fault finding effort by adding additional information to the email, such as what you were attempting to do within Sojour when it crashed.

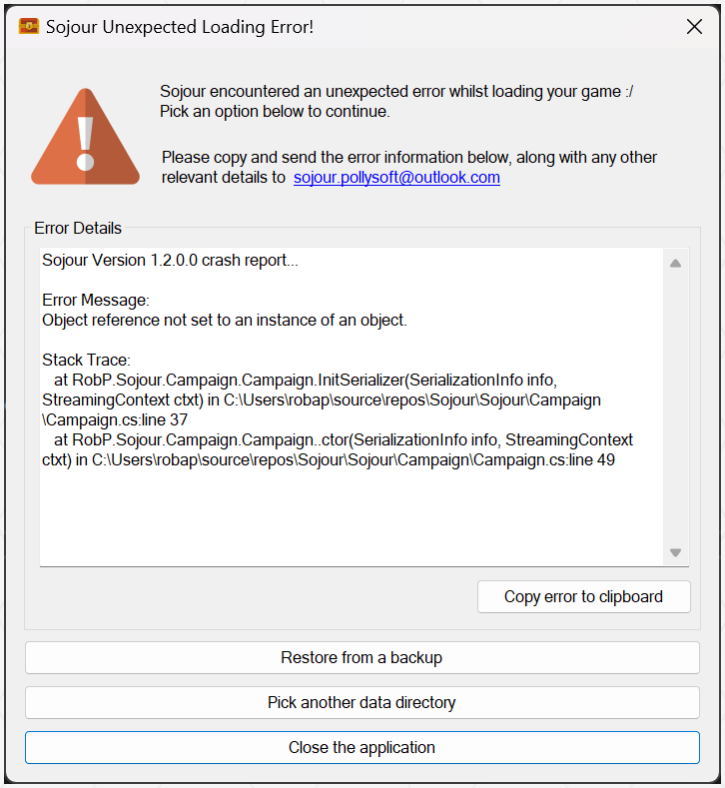
To exit this window click the **Save and Close the Application** button. This will have Sojour perform a save then close down.

All saved files will have a backup made of the file prior to the save being made. This is due to the fact that some crashes can corrupt the system so badly that the save itself could be corrupted. Sojour is designed to detect corrupt saves and to use the backups where these are detected.

If Sojour is forced to use a backup, you could lose some work. It is why I recommend clicking the save button on your journals at regular intervals, just to make sure your work is saved to disk.

Crashing During Loading

If Sojour crashes whilst loading, a slightly different screen will be displayed to the user:



This screen offers the user the opportunity to:

1. **Restore from a backup:** This allows the user to restore the current corrupt saved game from a backup. Note, this backup WILL over-write your existing corrupt saved game (a warning will be presented to the used)
2. **Pick another data directory:** This option allows the user to use a different data directory for their saved game. The existing broken saved game and its directory will be untouched.
3. **Close down Sojour:** Sojour will be close down and exit.

The rest of the functionality on this dialog window is identical to that on the general exception handler window described in the previous section.