# Chapter 2: Introduction to Altair Analytics Workbench

## Introduction

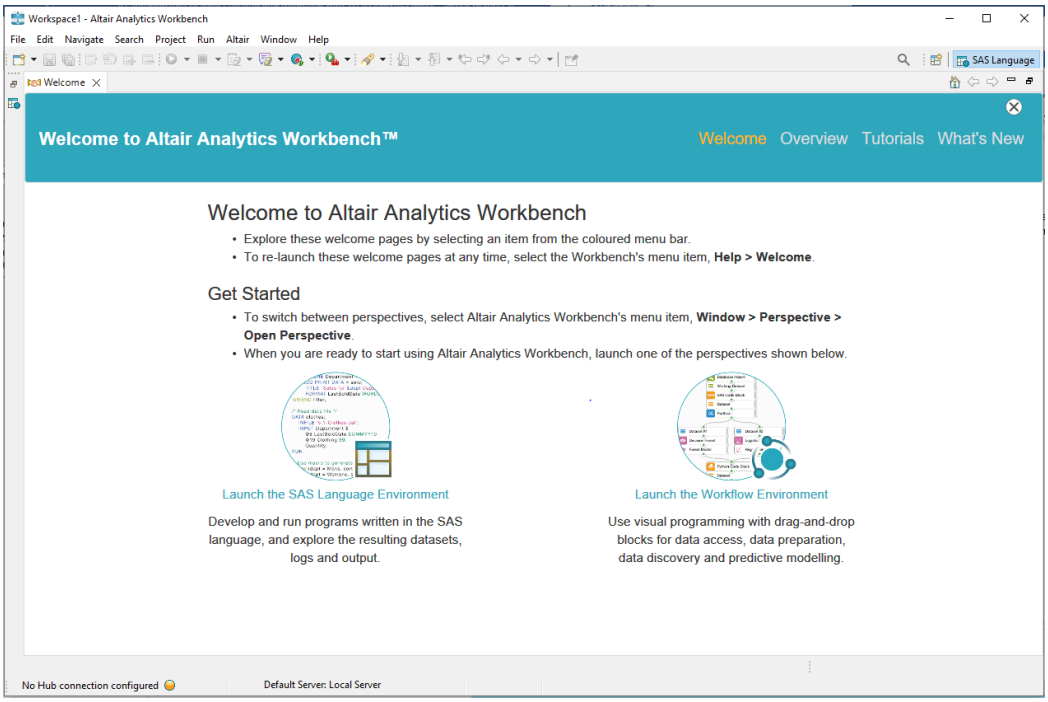
Contents for this lesson include an introduction, a look at Altair Analytics Workbench, the SAS language and workflow perspectives and common elements. We'll then speak about preferences and help options prior to getting onto to a demonstration and then returning to slides to highlight important points before wrapping up with a summary.

- Introduction
- Two perspectives
- SAS Language
- Workflow
- Common perspective elements
- Preferences
- Help
- Demonstration
- Things to remember

## What is Altair Analytics Workbench

So what is Altair Analytics Workbench? its a data processing tool designed to work with and for SAS programmers but also incorporates the capability to code in SQL, R and Python. The ability to create processing chains using drag and drop functionality is also available via Workflow.

Figure 1: Altair Analytics Workbench



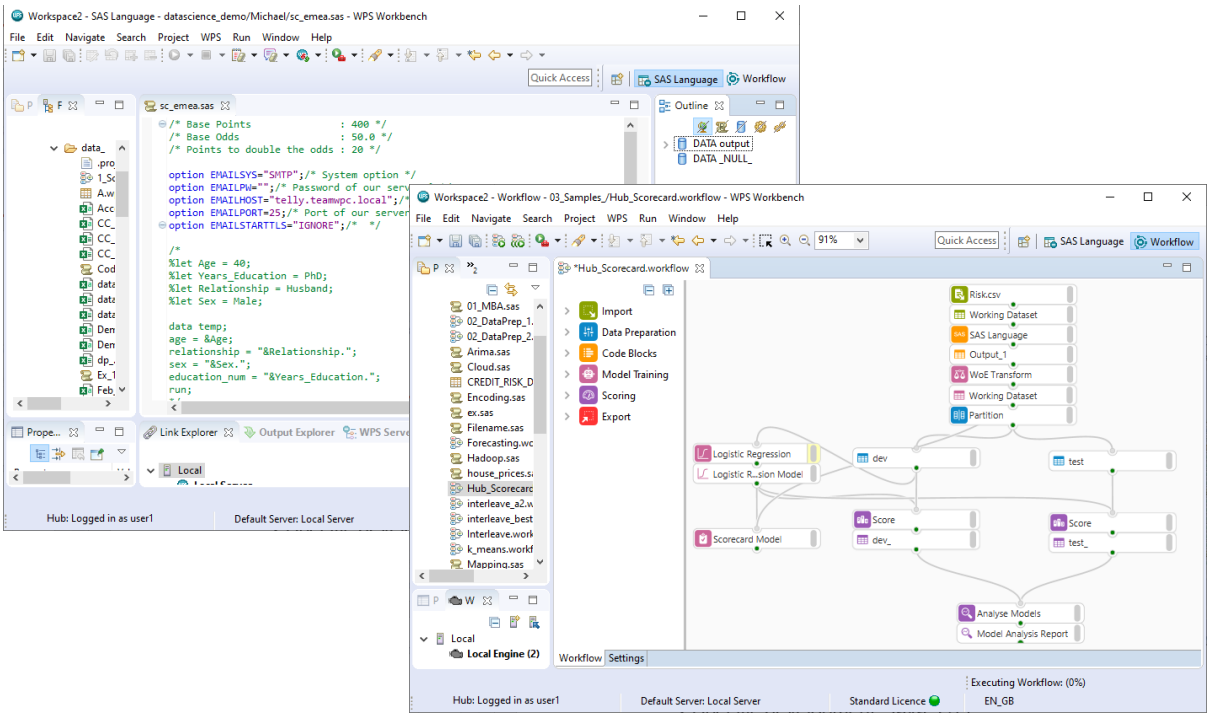

Altair Analytics Workbench is a customized eclipse environment. As a result of this some features such as menus and options are not used.

## Two Perspectives

Altair Analytics Workbench comprises two perspectives: the SAS Language perspective and the Workflow perspective.

The SAS Language perspective provides a fully equipped integrated developer environment for coding in the languages of SAS, R, Python and SQL and the Workflow perspective enables visual programming via drag and drop functionality with point and click wizards to configure.

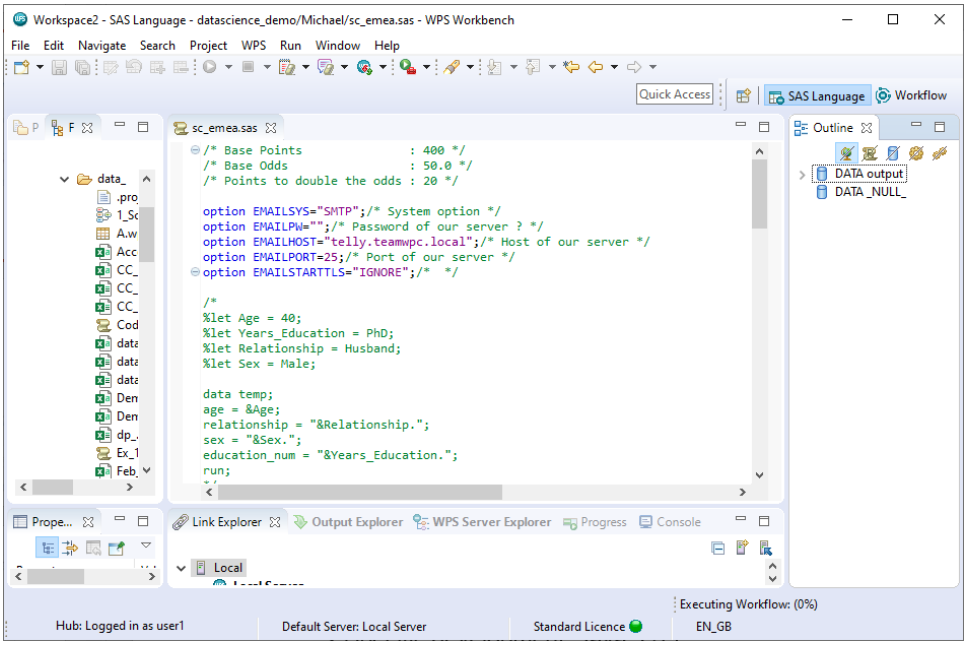
Figure 2: SAS Language and Workflow Perspectives



## SAS Language perspective

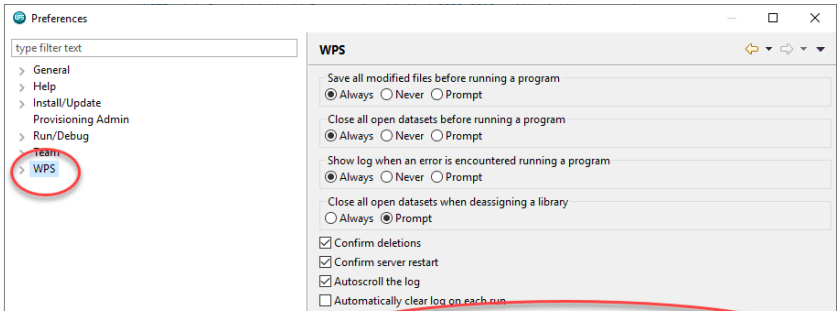
As stated, the SAS language perspective provides the ability to code in the languages of SAS, Python, R and SQL. Code assistance features are available including drag and drop options for filenames and variables, as well as providing a log for error tracking, verbose code and computation time.

Figure 3: SAS Language perspective




Output is sent to the Output Explorer and all session data is processed in a temporary location called the Work folder. The Work folder is destroyed when the program is closed and is an important aspect to bear in mind when using the SAS Language perspective. The location of the work folder can be viewed and set from Window > Preferences.

Figure 4: Window ... Preferences

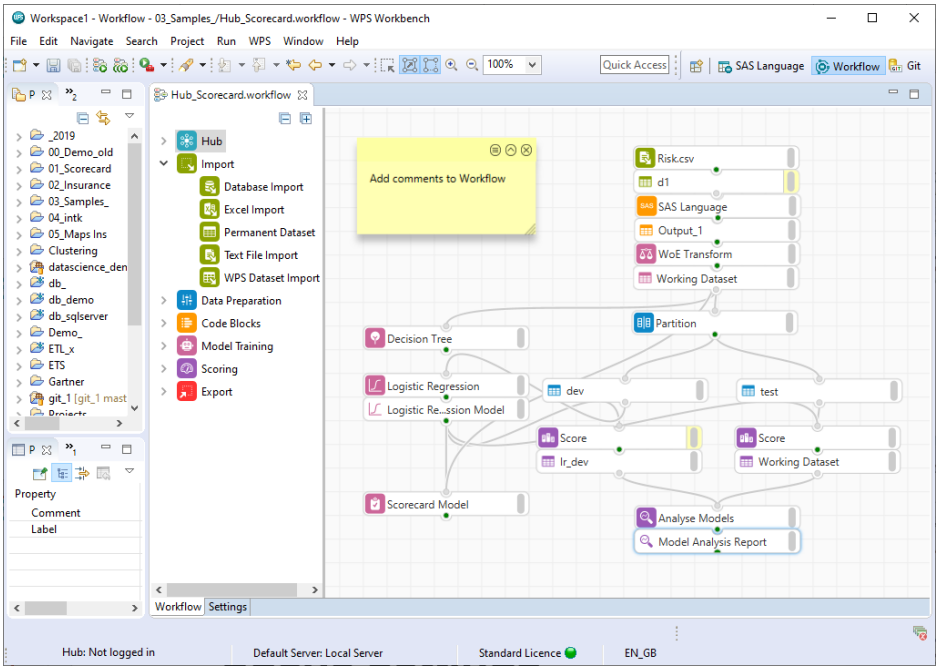


The SAS Language perspective also uses servers to process code and many servers can be configured.

## Workflow perspective

The Workflow perspective provides the ability to design processing chains using drag and drop and point and click functionality via blocks which are stored in groups, for example the Import group contains import blocks.

Figure 5: Workflow perspective



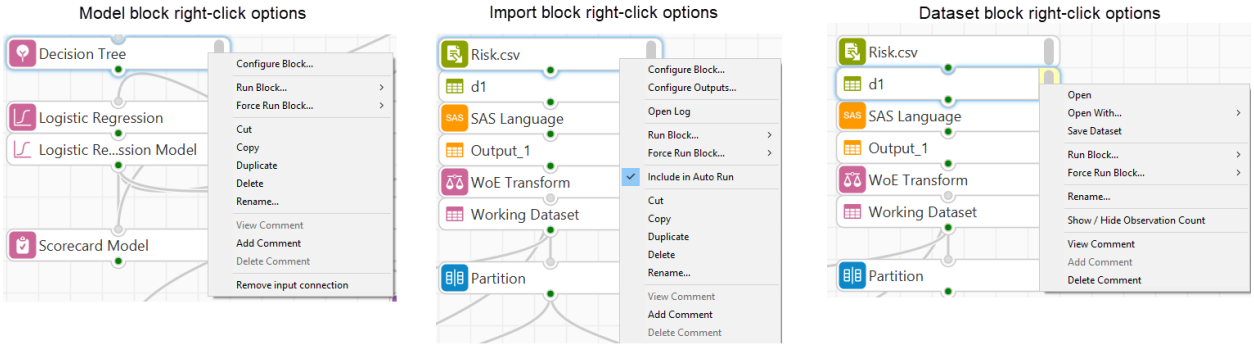

Once dragged from a group and placed on the Workflow, right-click any block for options. Options may differ depending on the type of block selected.

For example, right-clicking a model block includes options to configure, run, edit functions are available and the ability to include comments is also present.

Right-clicking a block that generates output, such as an import block provides options to configure the block and its outputs, access the block log, edit and add comments.

The dataset block includes options to open, save, run, rename, show/hide observation count and add comments.

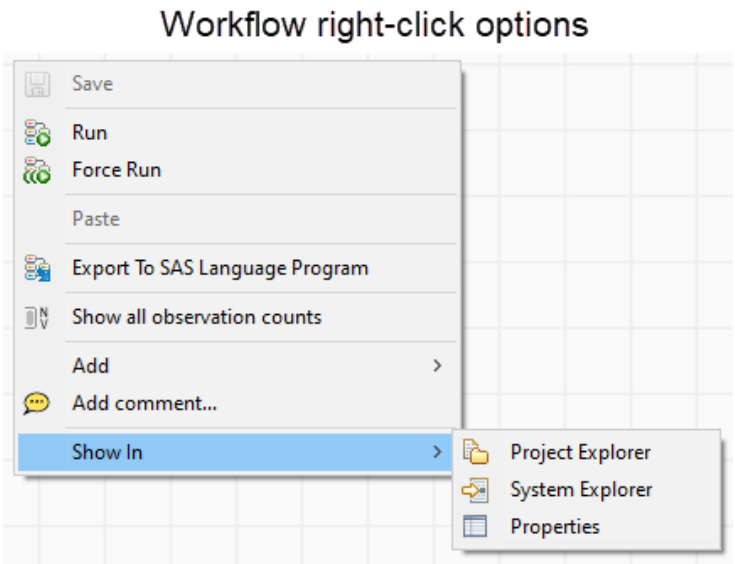
Figure 6: Right-click block options




Options are also available when the canvas is right-clicked. Right-clicking the Workflow canvas provides options to run the Workflow and export the entirety of a Workflow to a SAS language program.

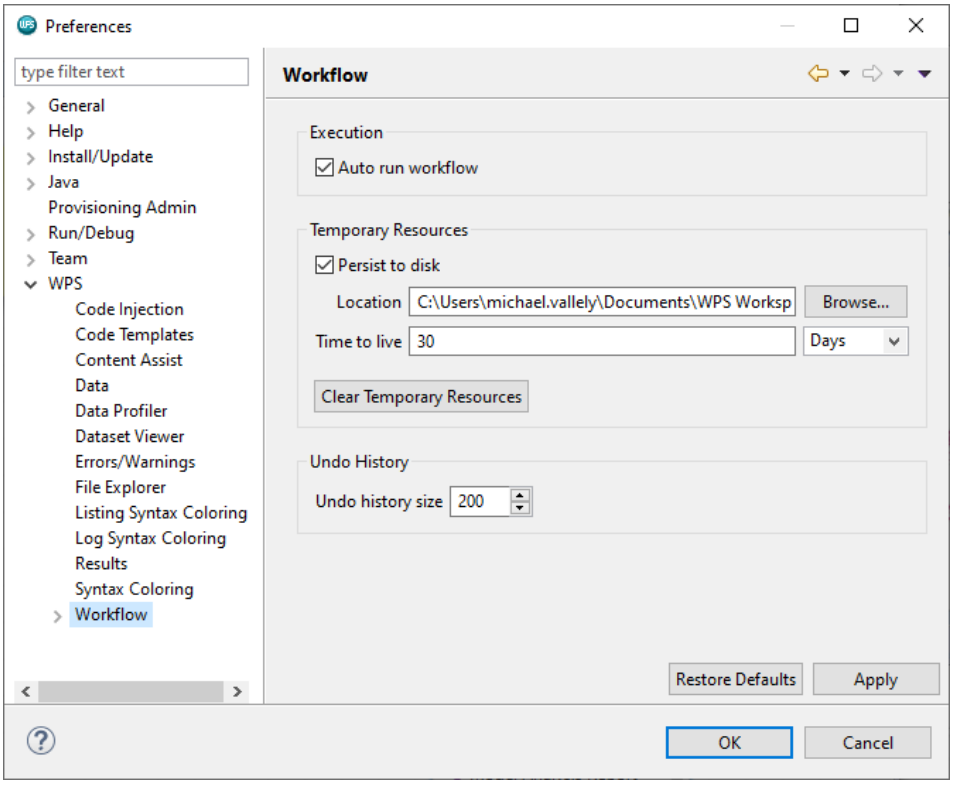
There is an option to show observations counts for all Workflow datasets, comments can be added and the Workflow located in Project Explorer, System Explorer or the Properties view.

Figure 7: Block and Workflow right-click options



The Workflow perspective does not include a Work folder. It retains the Workflow for a set period in a cache, the cache location and time to retain can be changed by modifying cache options from Window > Preferences > ALTAIR > Workflow.

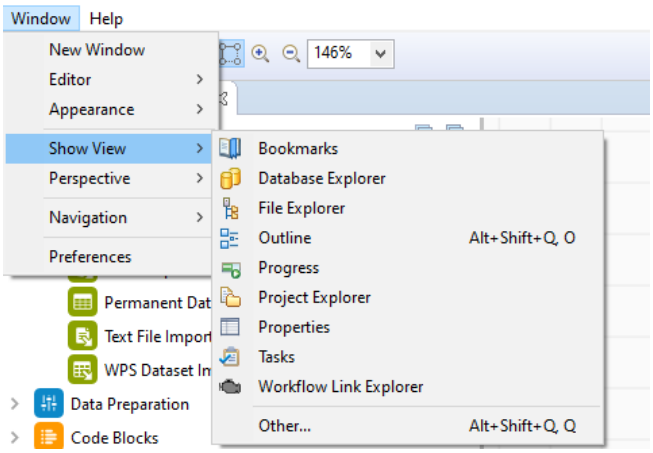
Figure 8: Workflow cache options




## Common perspective elements

There are many windows and views available in the Workbench and theses can be accessed and opened from within a perspective from Window > Show View.

*Figure 9: Accessing views*



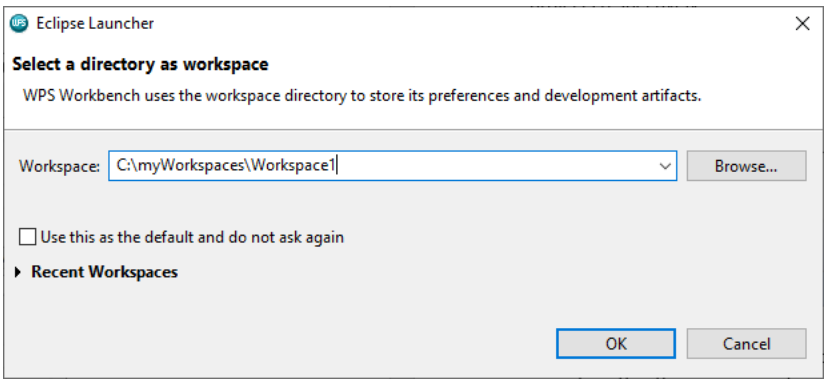

There are some common elements and windows evident across perspectives. Menus are evident with many options, the most used include File, Project and Window.

The File Explorer and Project Explorer provide the ability to navigate a file system and create and manage projects respectively and the facility to navigate perspectives is available by clicking the appropriate icon in the top right-hand corner.

Figure 10: Navigate perspectives


Another common element that is not explicit is a Workspace. Regardless of the perspective chosen to operate within this is a required aspect and is chosen at start up. A workspace is a location, any location, that stores session settings and many Workspaces can exist.

Figure 11:Setting a Workspace at start-up

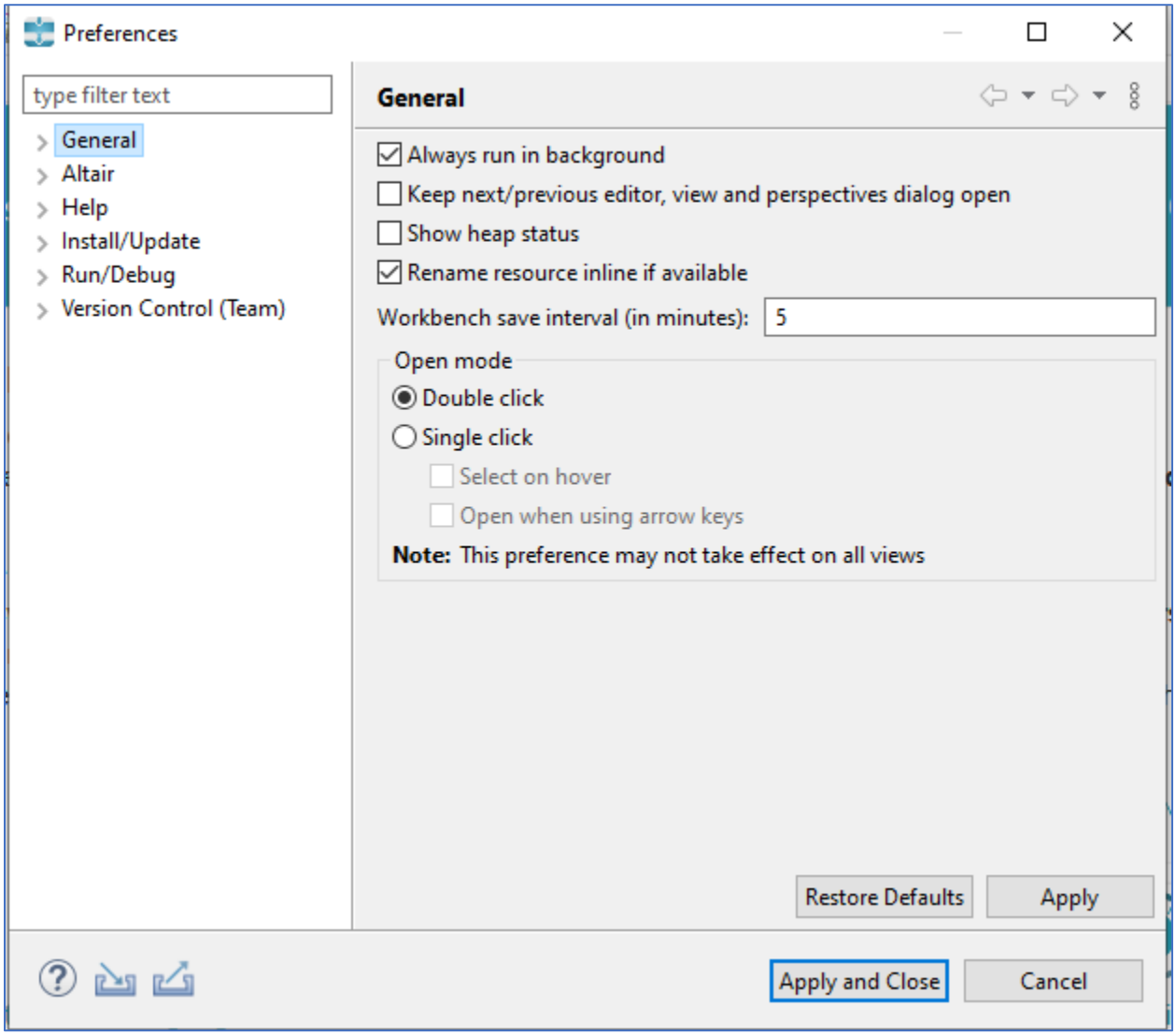


## Preferences

The menu item Window > Preferences provides the facility to set everything! There are a lot of options and as stated previously, some Eclipse features are not used and these include:

- Help,
- Install/update
- Provisioning Admin
- Run/Debug
- Team

Figure 12: Window .... Preferences

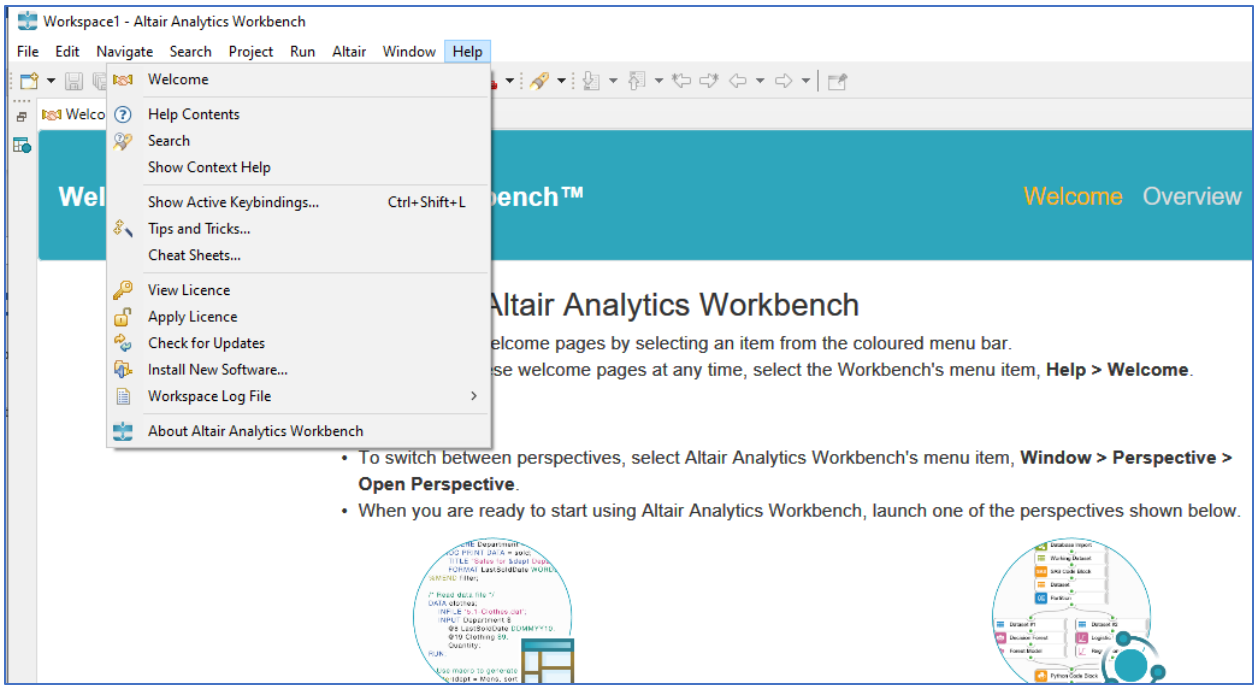


Useful preferences are contained in the General and ALTAIR options. General provides options to specify Workbench and interface settings and ALTAIR provides SAS Language and Workflow perspective options.

## Help

The Help menu contains help options. Options available include the Welcome screen that pops up when the program is first run to in-depth help files, contextual help, tips and tricks and cheats sheets.

*Figure 13: Help options*

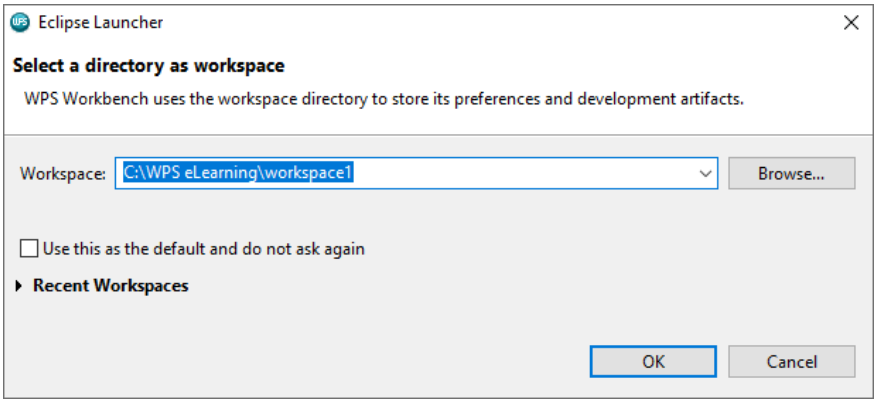



The help menu also contains license and product information and a shortcut to keyboard shortcut settings via the option: Show Active Keybindings…

## Demonstration

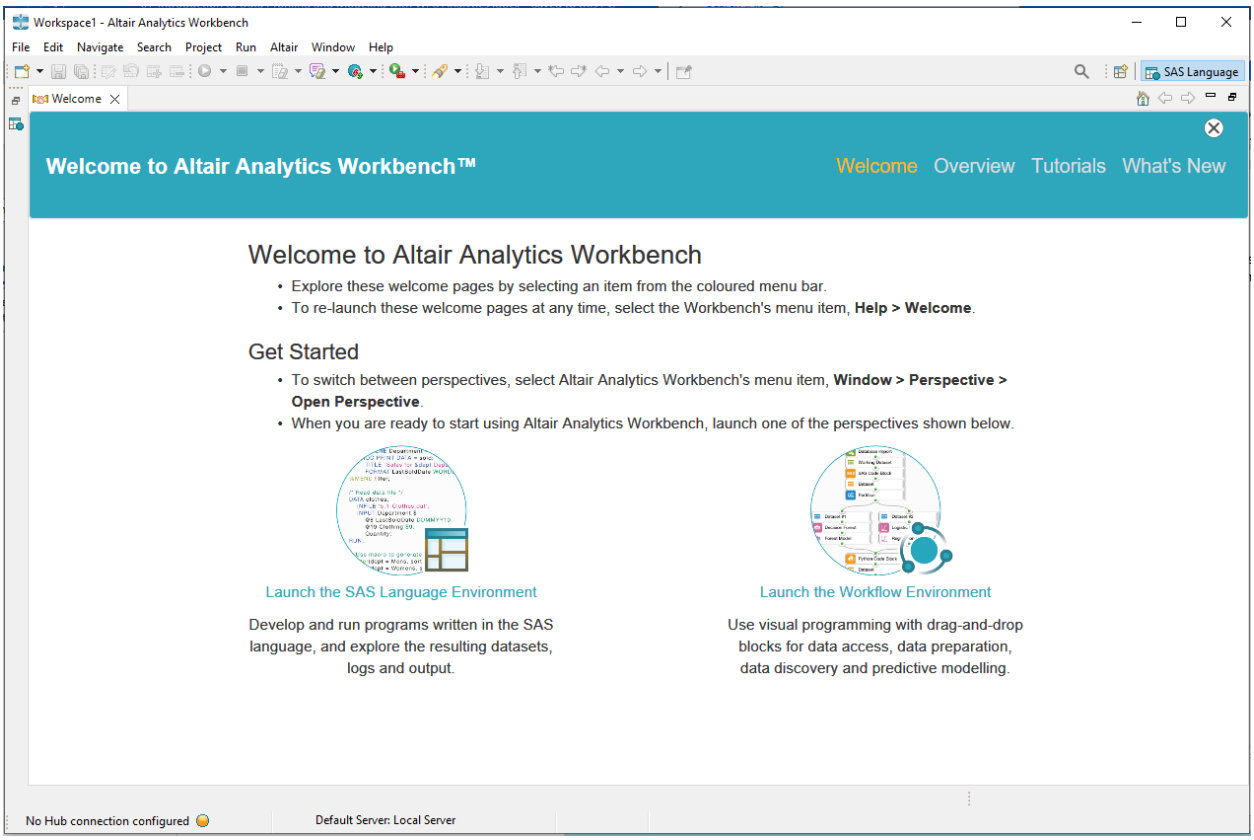
So let's get on with a demonstration. The first thing to do is start the program, on first start a Workspace must be set and this can be any folder with any name in any location. The Workspace stores all sessions settings and bear in mind that multiple workspaces can exist.

Figure 14: Setting a Workspace




When setting the Workspace always point to a known and accessible location, here the Workspace is set to a folder called *workbenchworkspace* in a folder called *ALTAIR eLearning* in the local *c\:drive.* The program opens at the welcome screen which is split into three sections.

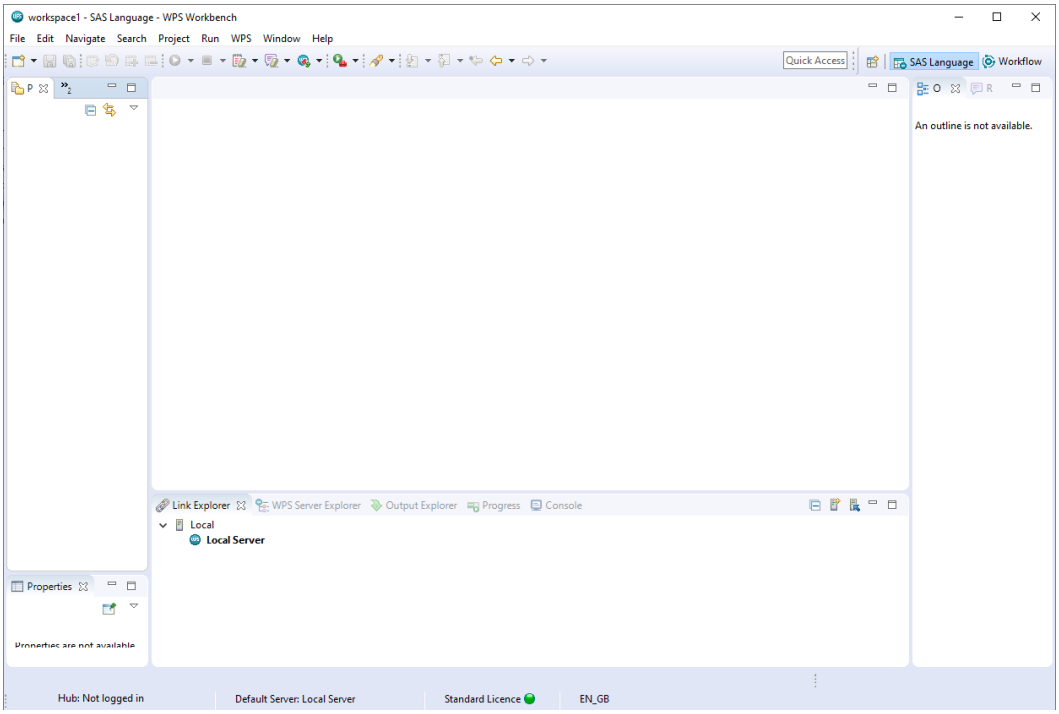
Figure 15: Welcome




Help options are available from the menu bar including links to an Overview, Tutorials and What's New. The steps to return to the welcome screen via help menu option are relayed in the central section along with Information on how to switch between perspectives with quick launch links to open either the SAS Language perspective or the Workflow perspective.

Rather than choose one of the options to launch either the SAS Language or Workflow perspective, the Welcome screen is closed, and the SAS Language perspective is visible.

Figure 16: SAS Language perspective

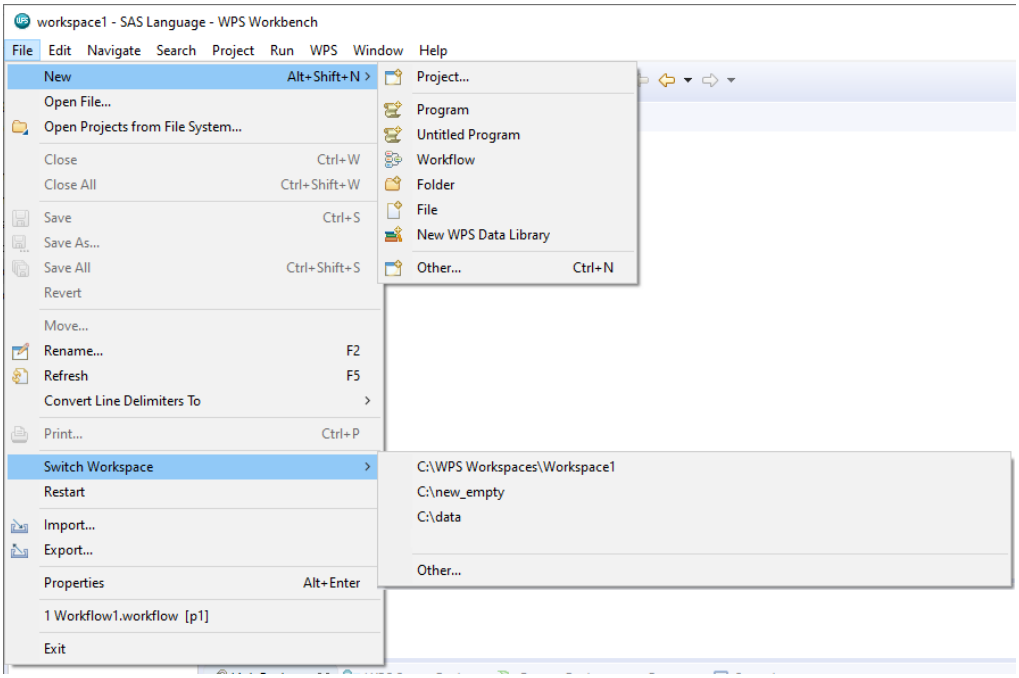



As has been stated the Workbench comprises two perspectives, the SAS Language perspective and the Workflow perspective. Each perspective comprises a set of views that can be moved, docked, closed and configured to suit needs.

The menu system and some views such as the Project Explorer, File Explorer, Database Explorer and Properties view are visible by default across both perspectives.
Some views are perspective specific. For example, the Link Explorer , the outline and
Results explorer views are specific to the SAS Language perspective. Although they can be open in the Workflow perspective, are redundant to it.

The menu system provides familiar options accessible across perspectives and the taskbar provides shortcuts for commonly used menu options, hover for tool tip.

Figure 17: File menu options




**The File menu** provides options to create new projects, programs, Workflows, folders, files and libraries or open already existing ones.

Notice also the option to Switch Workspace to either a previously existing one or a new one via Other…

**The Edit menu** in general contains generic edit options.
**The Navigate, and Search menus** provide the facility to locate and navigate resources and other artifacts displayed in the Workbench.

**The Project menu** provides project options including access to selected project properties.
**The Run** menu enables configuration of third-party tools that can integrate with the Eclipse environment.

**The ALTAIR menu** provides the ability to login to Altair SmartWorks Analytics Hub; a web-based facility which includes user and data access governance with program deployment capabilities. This menu also includes run, log and server and link options.

Figure 18: ALTAIR menu

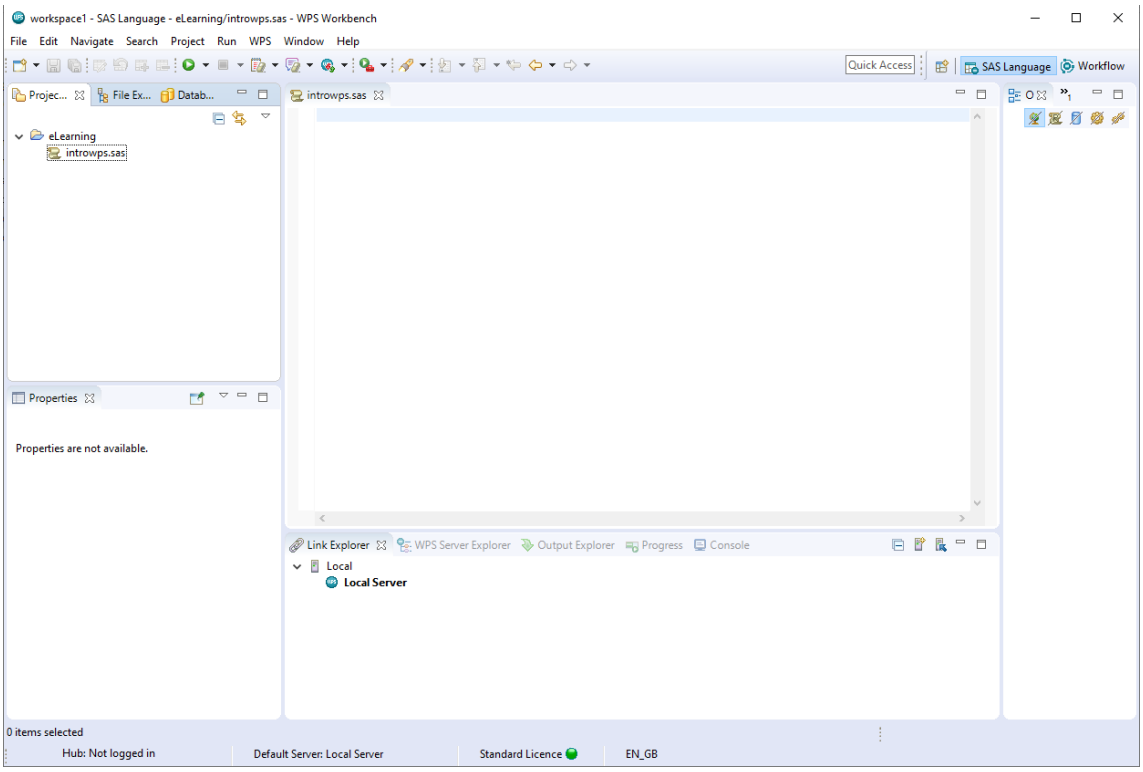
**The window menu** provides options to split the screen, show views and switch perspectives as well as providing access to system Preferences, more on this later.

Figure 19: Window menu

# Coding in the SAS Language perspective

To begin coding a new program file must be added. To add a new program: the project folder is right-clicked and New > Program selected. A name is assigned, here *introAltair.sas.* Finish is clicked and the file appears in the project folder and opens in the code view.

*Figure 20:New program creation*

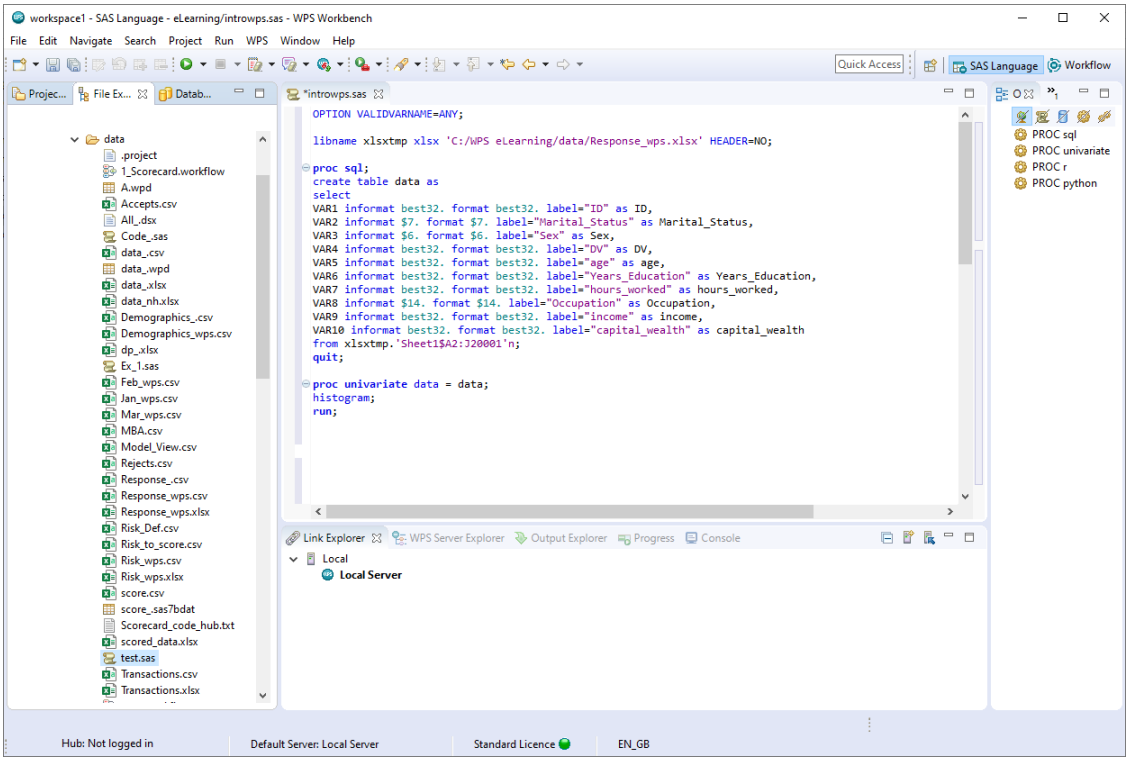



Code can be manually entered and run using this view but here some code will be copied from a file located using the File Explorer view.

The File Explorer can be used to navigate a file system and select and open files. For example, clicking a known file format opens the file in a new pane, selecting a proprietary file format, for example, a Microsoft Excel file will open in Microsoft Excel.

Here a code file *test.sas*, located in the eLearning data folder is selected and opened, the code file could be saved to the project folder, but here the code is copied and pasted to the file created earlier. All other files are closed.

Figure 21: Code pasted from *test.sas*



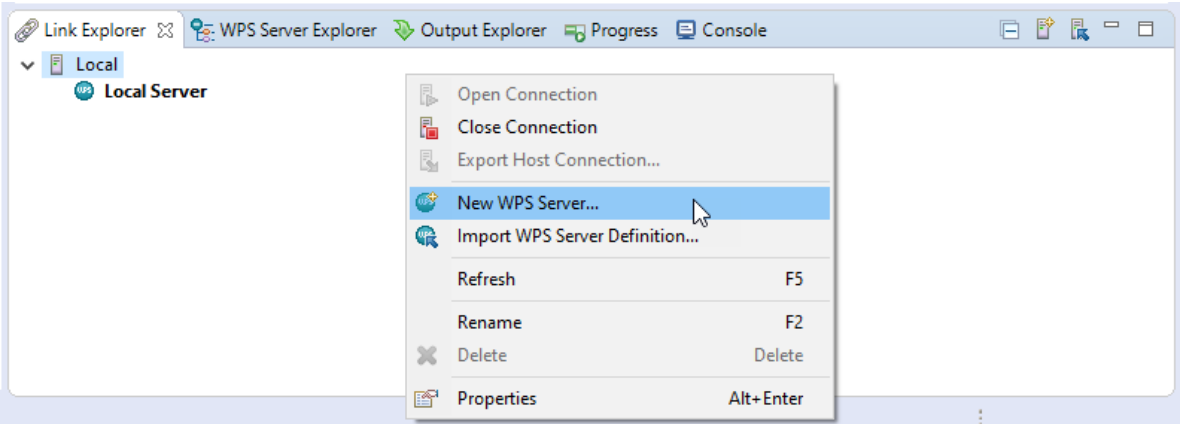

The code reads the file *response.xlsx* from the eLearning data directory and runs some analysis on its contents. In order to run a program written in the language of SAS, Altair Analytics Workbench requires a connection to an ALTAIR server.

The term server here is used to refer to a server process. The process may be running on the local workstation machine - a local host connection, or on an installation of ALTAIR on a remote machine - a remote host connection.

Servers running under the local connection are termed local servers and servers running under a remote connection are termed remote servers.

With a full workstation installation of ALTAIR there will be a single host connection called Local and a server called Local Server.  This server will be started by default when Altair Analytics Workbench is started.

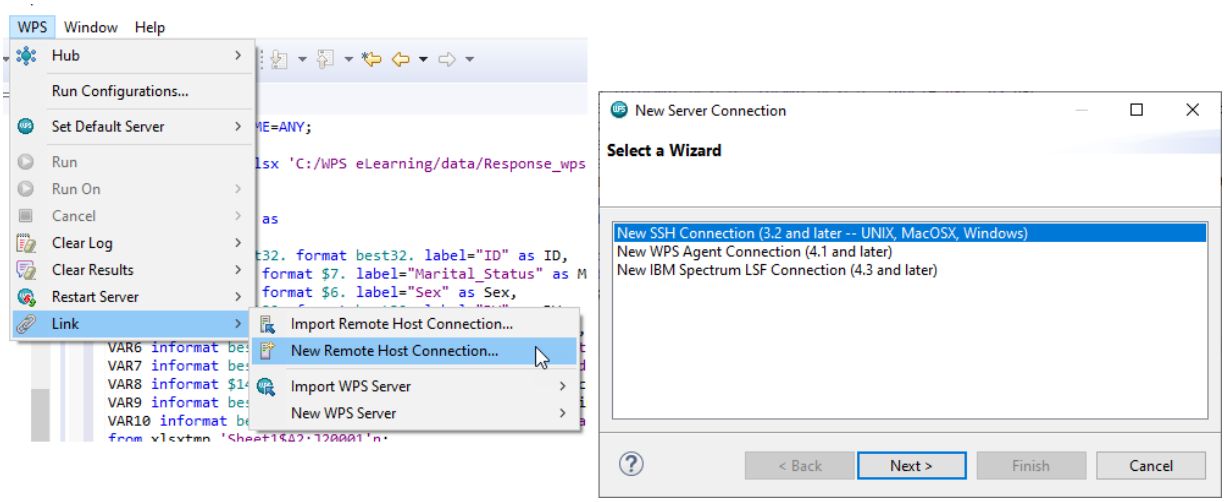
*Figure 22: Local Server*




Additional local servers can be configured by right-clicking Local from within the Link Explorer view and choosing the option New ALTAIR Server, this will create a new local connection.

A remote server connection, which is used to configure a thin client to a server, can be added from the ALTAIR menu by selecting Link and then New Remote Host connection. Server details will need to be supplied to create the connection.

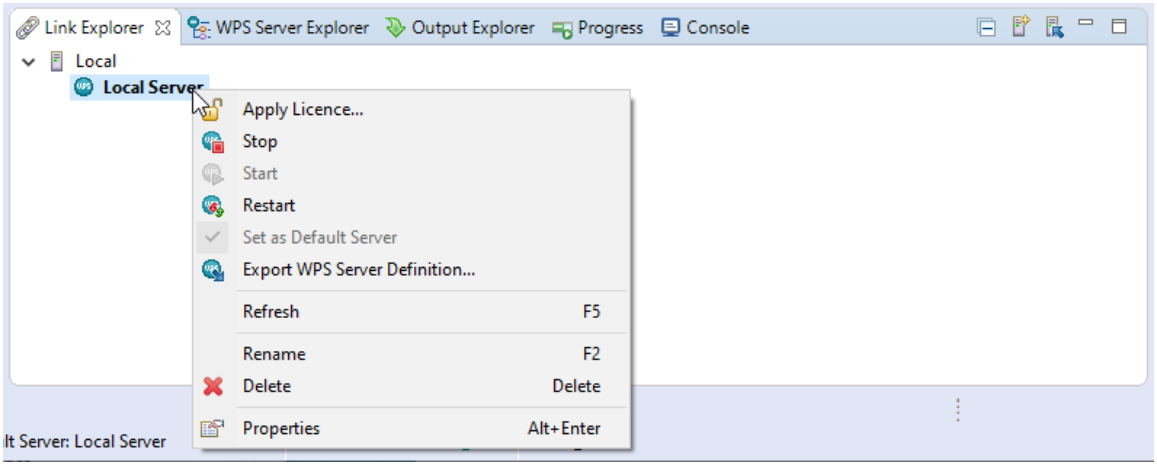
Figure 23: New Remote Host Connection...




Regardless of whether a local or remote connection is evident, the Workbench functionality, look and feel remain the same.

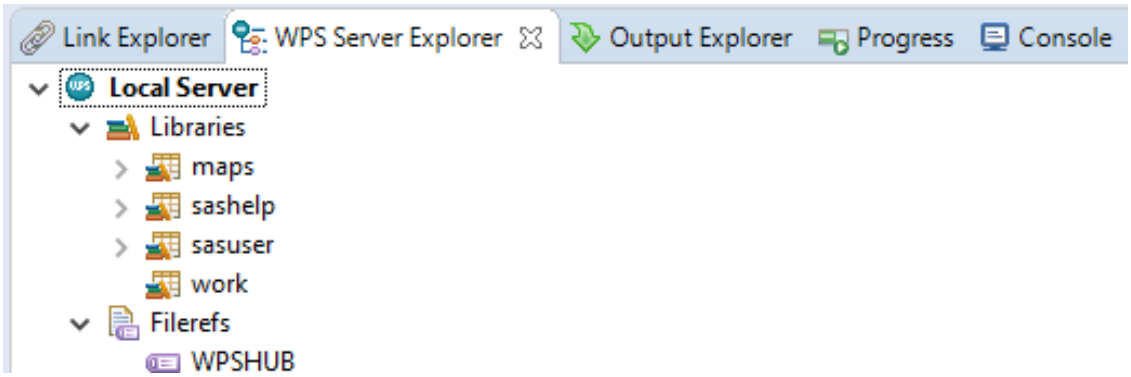
The Link Explorer tab displays all hosts and servers connected to those hosts. Here this is only one - on the local machine and this will process all code, right click for options including properties.

Figure 24: Local Server options




The ALTAIR Server Explorer view is simply that: a view of the contents of connected servers, in this instance the local server.

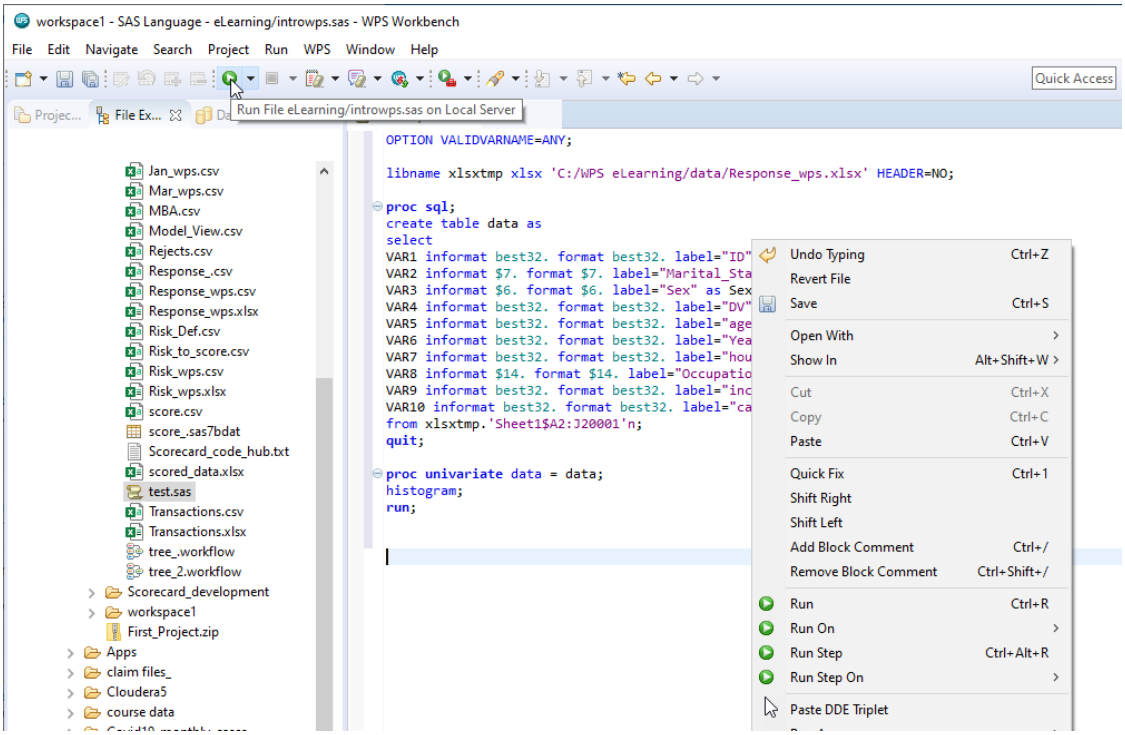
Figure 25: ALTAIR Server Explorer




This view displays a List of Filerefs, libraries, catalogs and datasets generated from running a SAS language program on a ALTAIR server. Notice the work folder, this is the default temporary location that houses all data and is flushed when the program is closed.

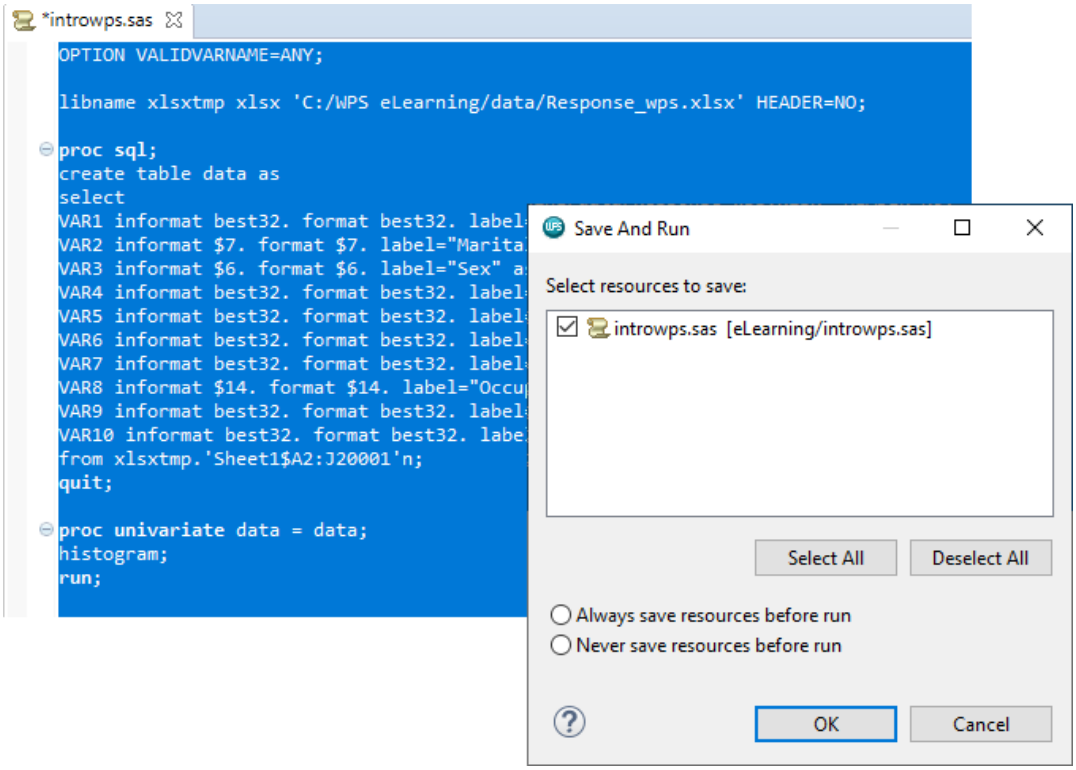
The code in a file can be run by selecting the run option on the task bar or by right clicking anywhere in the code view and selecting an appropriate option or by using the keyboard shortcut CTRL+R.

Figure 26: Run options




Code can be run in full or partially by selecting the code to run and choosing a run option. Here this code to import the file and name it: **data** is selected and CTRL+R is used to run the code.

Figure 27: Save and Run

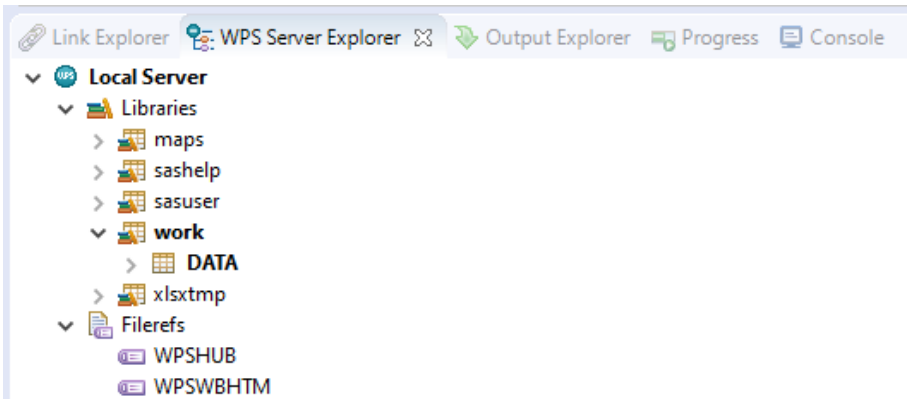



A message appears when running code for the first time, the message relays information about the file save status. Note the asterisk - meaning the file has been modified but not yet saved.

Selecting either *Always save resources before run* or *Never save resources before run* will ensure this dialog never appears again. A similar dialog will appear prompting in relation to closing data files that are referenced in code that are currently open.

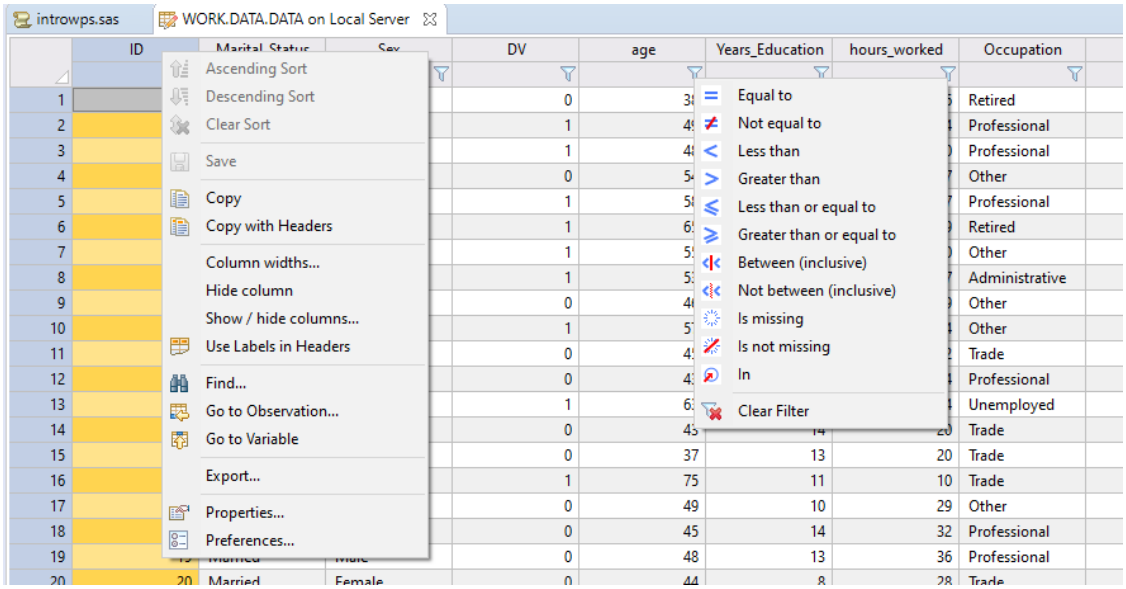
Clicking OK saves the file is saved, note the asterisk has disappeared, the code runs and a file is output to the work directory.

Figure 28: Work directory



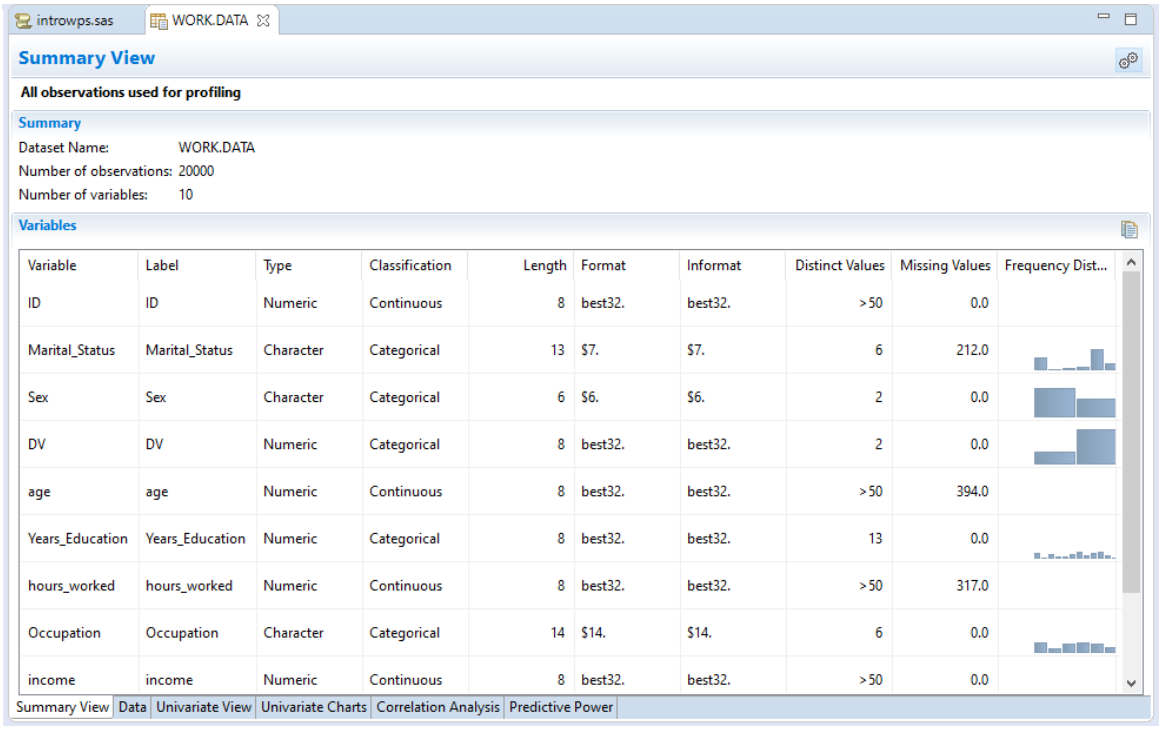
The file can be accessed in a number of ways, doubling clicking opens the data in the Dataset File Viewer: this provides a spreadsheet style display with sort, edit and filtering options available.

Figure 29: Dataset File Viewer




Returning to the Work folder and right-clicking the dataset and selecting Open With provides options to open the data with the Dataset File Viewer as just illustrated, or the Data Profiler.

Figure 30: Data Profiler

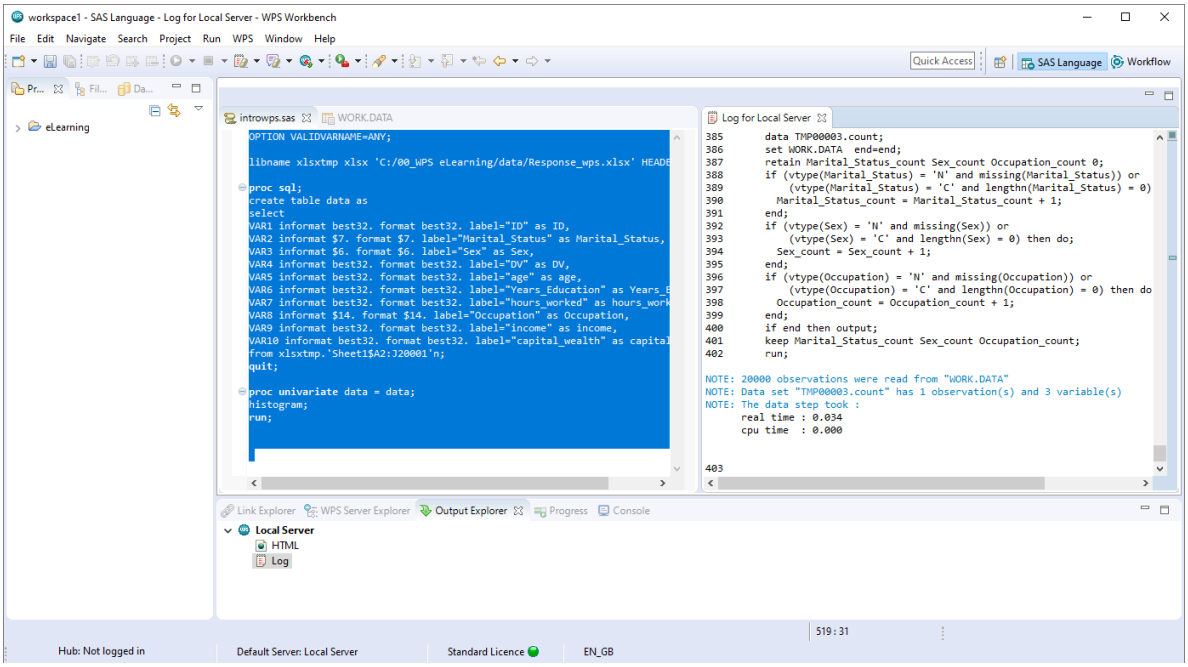



| Variable | Label | Type | Classification | Length | Format | Informat | Distinct Values | Missing Values | Frequency Dist... |
|---|---|---|---|---|---|---|---|---|---|
| ID | ID | Numeric | Continuous | 8 | best32. | best32. | >50 | 0.0 | |
| Marital_Status | Marital_Status | Character | Categorical | 13 | $7. | $7. | 6 | 212.0 | |
| Sex | Sex | Character | Categorical | 6 | $6. | $6. | 2 | 0.0 | |
| DV | DV | Numeric | Categorical | 8 | best32. | best32. | 2 | 0.0 | |
| age | age | Numeric | Continuous | 8 | best32. | best32. | >50 | 394.0 | |
| Years_Education | Years_Education | Numeric | Categorical | 8 | best32. | best32. | 13 | 0.0 | |
| hours_worked | hours_worked | Numeric | Continuous | 8 | best32. | best32. | >50 | 317.0 | |
| Occupation | Occupation | Character | Categorical | 14 | $14. | $14. | 6 | 0.0 | |
| income | income | Numeric | Continuous | 8 | best32. | best32. | >50 | 0.0 | |

The Data Profiler includes the Dataset File Viewer via the data tab, but also provides others profiling tabs for more in-depth assessment.

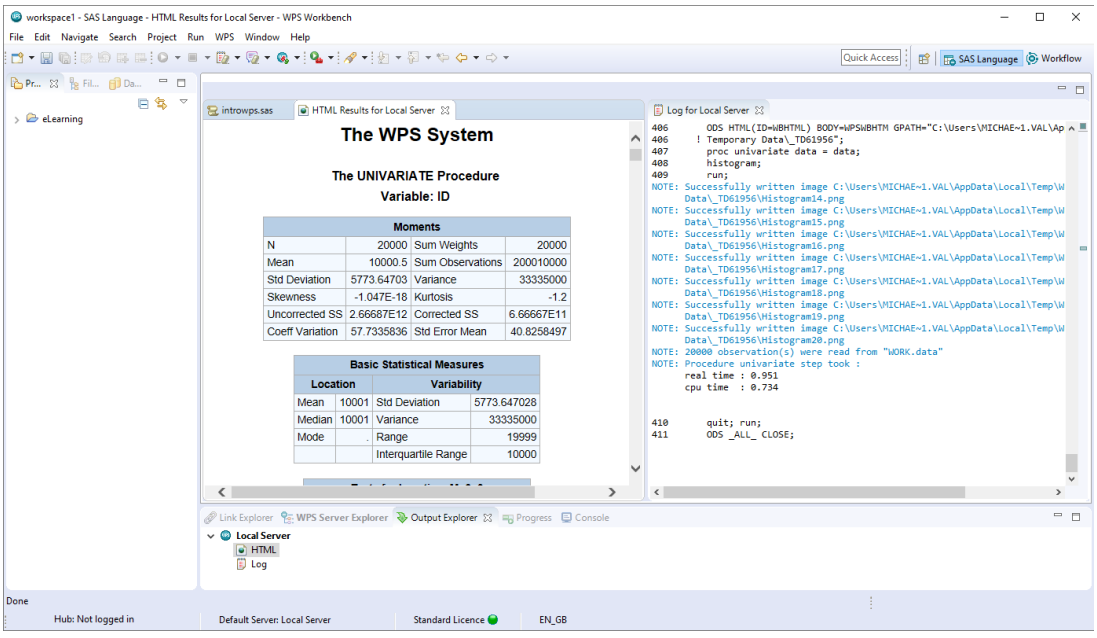
The Output Explorer provides access to logs and HTML output. Thankfully as screens can be split the log can be viewed side by side with code.

Figure 31: Output Explorer




The log contains verbose code, computation time and any warnings or errors. Notice the log populating as the remaining code is selected and run. This code analyses the file and outputs results. Clicking the HTML node opens the HTML Viewer with output generated available.

Figure 32: HTML Viewer




Note that the screen can be split further to accommodate more items.

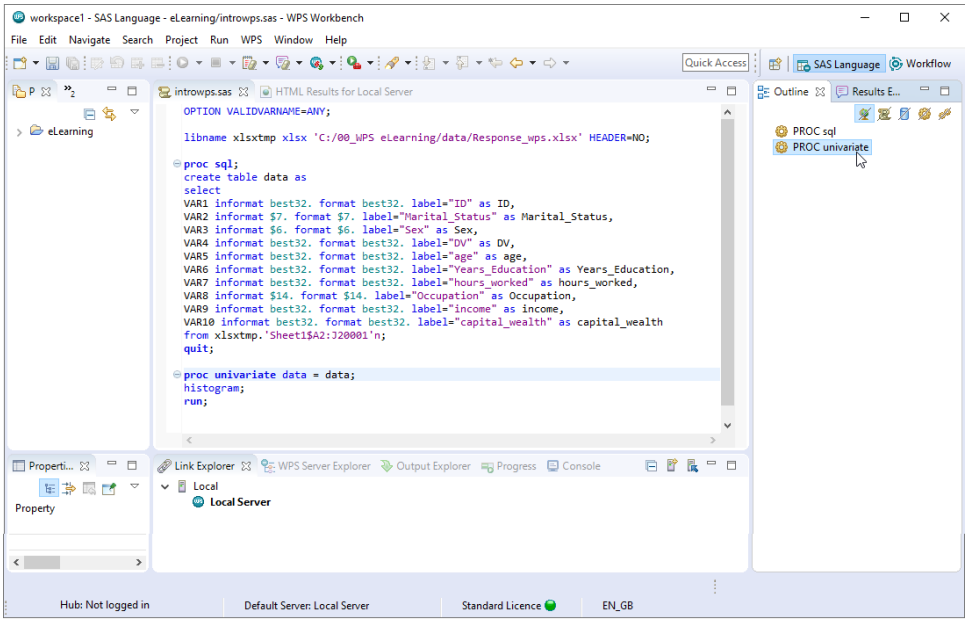
Code assistance features are available including the ability to list relevant elements via holding the CTRL key and hitting the space bar. Coloring coding and error tracking are also included.
Here the data set in the Work directory is duplicated and a new variable added. Notice the dataset and variable names can be dragged from the work directory to assist when building code.

Highlighting and running the code, the log scrolls and the data is created and evident in the Work directory.

The Progress tab displays messages relating to processing code from the Local Server, for example selecting a piece of code and running shows an execution bar scrolling. The console tab is similar however this displays messages produced by ALTAIR Server.

The Outline view and Results view to the right, provide the facility to navigate code elements and output. For example, selecting any code element from the outline view highlights it in the code view, correspondingly selecting any entry from the results view navigates to it in the HTML output.

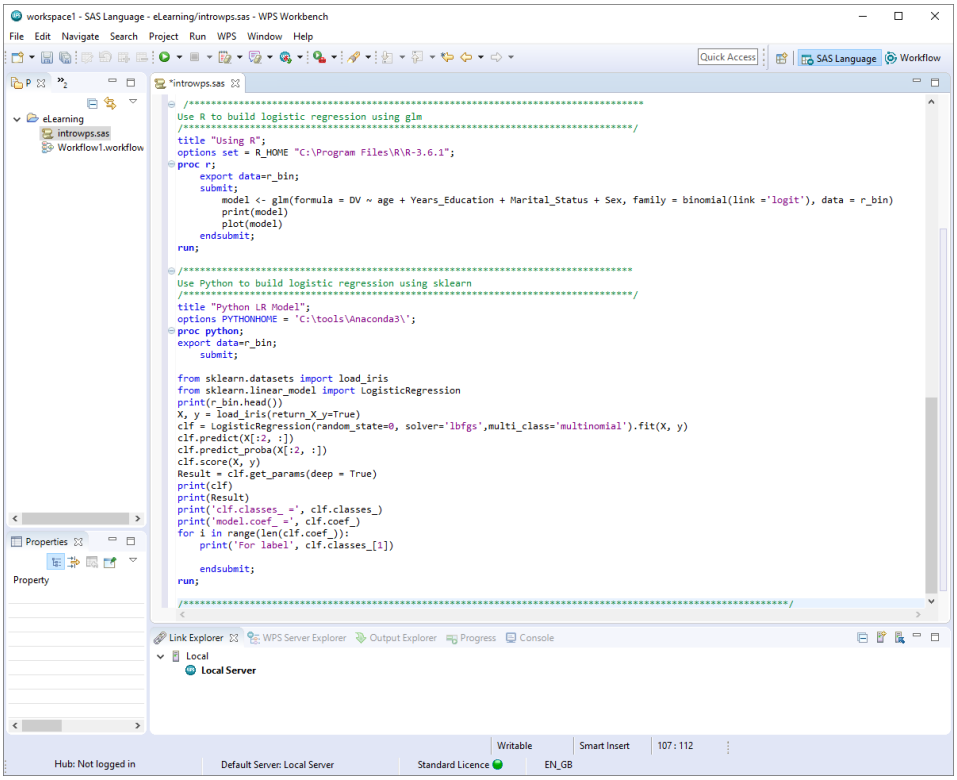
Figure 33: Outline view




The SAS Language perspective has many features for programming not only in the language of SAS but also for interleaving SQL, Python and R code

R and python code can be passed via the PROC R and PROC PYTHON wrapper procedures respectively. R or Python code is entered between the *submit* and *end submit* statement for each procedure.

Figure 34: PROC R and PROC PYTHON




Note that R and Python are neither installed or managed by Altair Analytics Workbench. R and Python can be installed either prior to or after installing Altair Analytics Workbench.

Altair Analytics Workbench enables R and Python code to be written in R and Python, the code is passed to the respective tools, processed and a response returned and displayed in the HTML viewer.

More information on using these languages can be found in the Help files.

## Workflow options and making connections

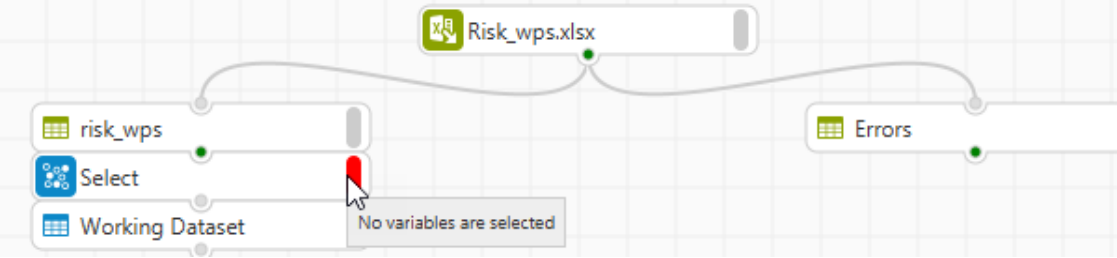
The output data can be detached from its source by clicking and dragging, this makes the connection visible.

Moving and panning around the Workflow canvas can be achieved by holding down the left mouse button while simultaneously moving the mouse. The middle mouse button controls zoom, this is also available from the task bar. Additionally, a grid and snapping features are also present.

Adding more blocks can be accomplished in three ways:

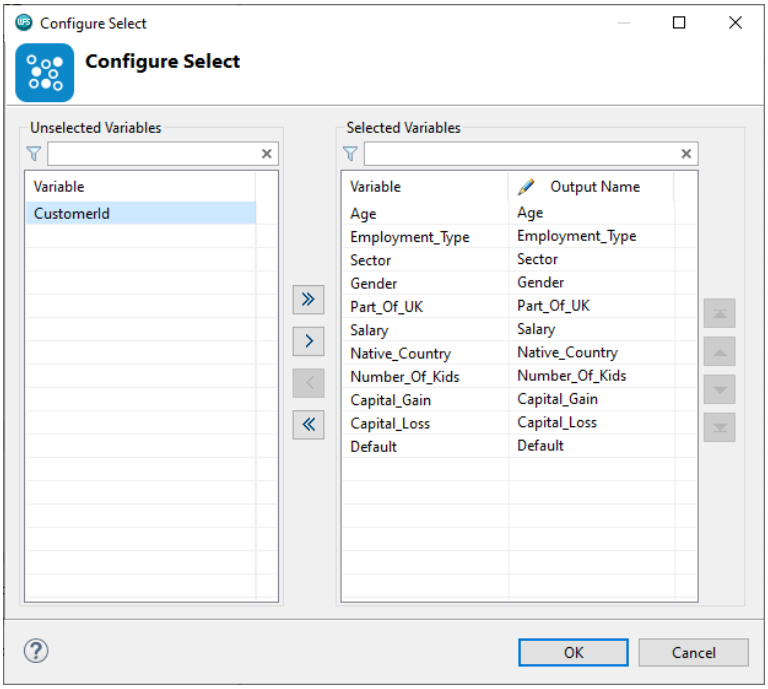
First, by dragging and dropping, here a Select block is dragged from the Data Preparation group to the dataset: *risk_Altair*, the block automatically docks and connects and the block indicator status shows that no variables have been selected.

Figure 35: Adding a Select block




Here, the block configuration dialog is accessed by double-clicking the block.

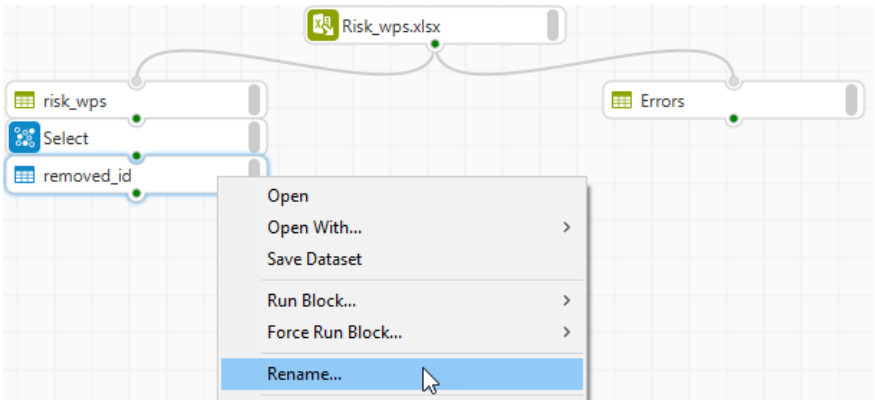
Figure 36: Configure Select




With the select block specific variables can be chosen for inclusion in a new dataset, here all variables bar *CustomerId* are moved from the unselected variables list to the selected variables list.

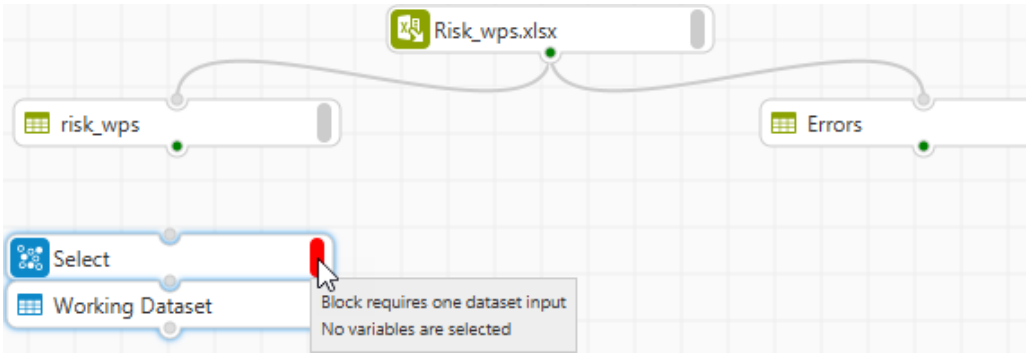
The process is executed by clicking OK. Notice the generic name assigned to the resulting dataset, here this is renamed to **removed_id**.

Figure 37: Rename dataset




The second method is similar but requires making a manual connection once the block is dragged onto the canvas. At this point, the block indicator status relays that the block requires one input connection and variables to be selected.

Figure 38: Manual connection

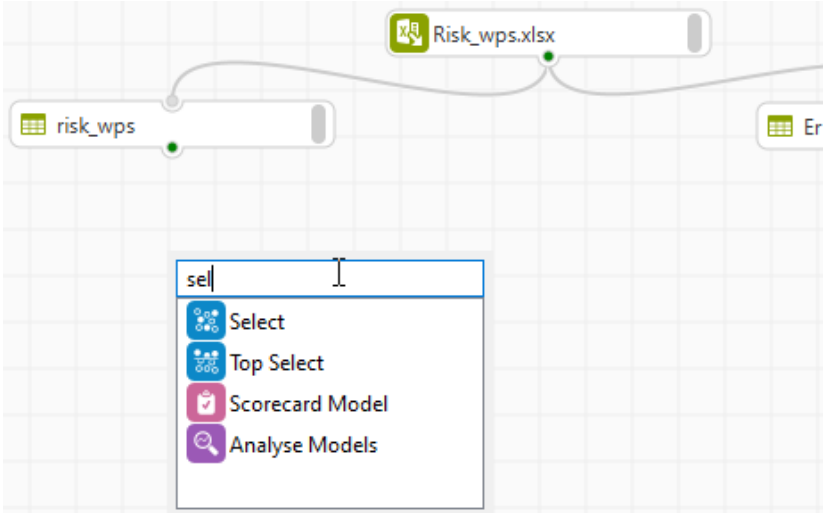



To connect the data to the block, the mouse is hovered over the output port of the dataset *risk_Altair*, the left mouse button is held and the mouse dragged and the connection released over the input port of the Select block.

The connection has been made and the indicator status for the select block has now only one message relating to configuration: No variables selected. Here the variable *age* is removed and the dataset renamed to *removed_age* to reflect.

A third way to add blocks is by double-clicking anywhere on the Workflow canvas. All blocks are listed as they appear in their respective groups.

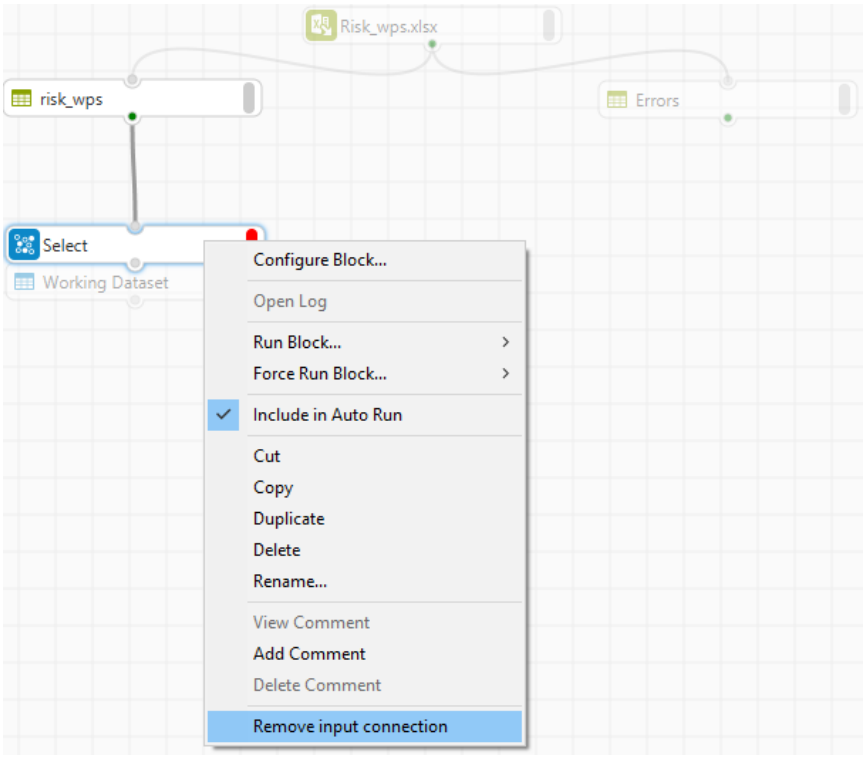
Figure 39: Adding blocks by double-clicking the canvas




Simply scroll to any block or use the search bar to automatically locate any block. Once located, double-clicking includes the block on the Workflow and it can be connected and configure as outlined previously.

Blocks are disconnected by right-clicking and selecting the option: Remove input connection.

Figure 40: Remove input connection

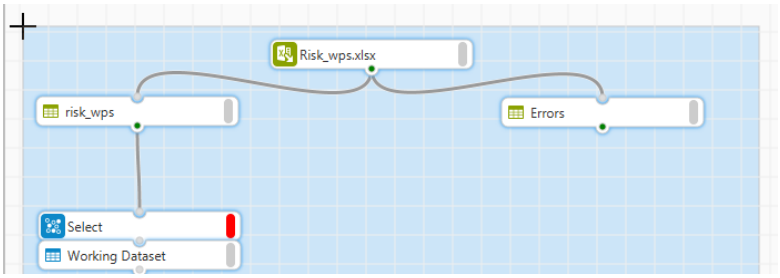



If multiple connections are evident then all are listed. Alternatively, the block can be easily deleted by clicking the *delete key* on the keyboard. Undo is also available by clicking CTRL+Z or by selecting Undo from the Edit menu.

Blocks and chains can be duplicated and pointed to new data. To select more than one block, the CTRL key on the keyboard is clicked and held until the crosshairs appear on the Workflow canvas, the left mouse button is clicked and the mouse dragged to select an area, any blocks included in that area are automatically selected.

Alternatively, blocks can be selected by holding down the CTRL key and selecting blocks individually.

*Figure 41: Selecting multiple blocks*

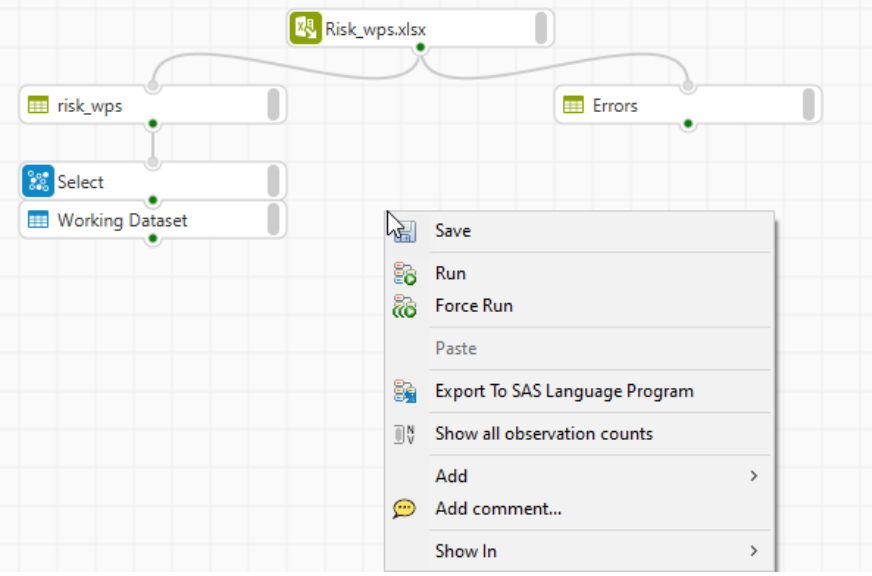


Using the keyboard shortcuts CTRL+C and CTRL+V, or identical options from the edit menu, the blocks can be copied and pasted.

This is useful when a large processing chain has been developed and needs to be applied to new data.

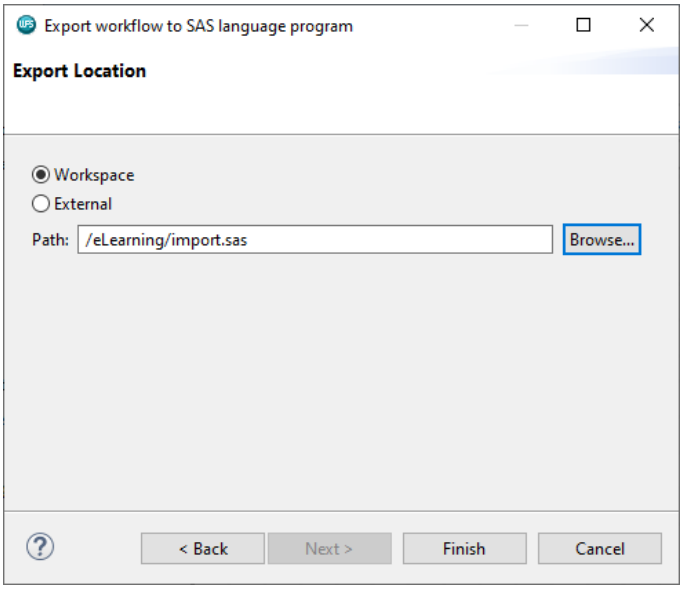
Right clicking the Workflow canvas provides options to add comments to the Workflow and export the entire Workflow to a SAS language program. Note that this option is only available if all Workflow blocks have a green execution status indicator.

Figure 42: Workflow canvas options



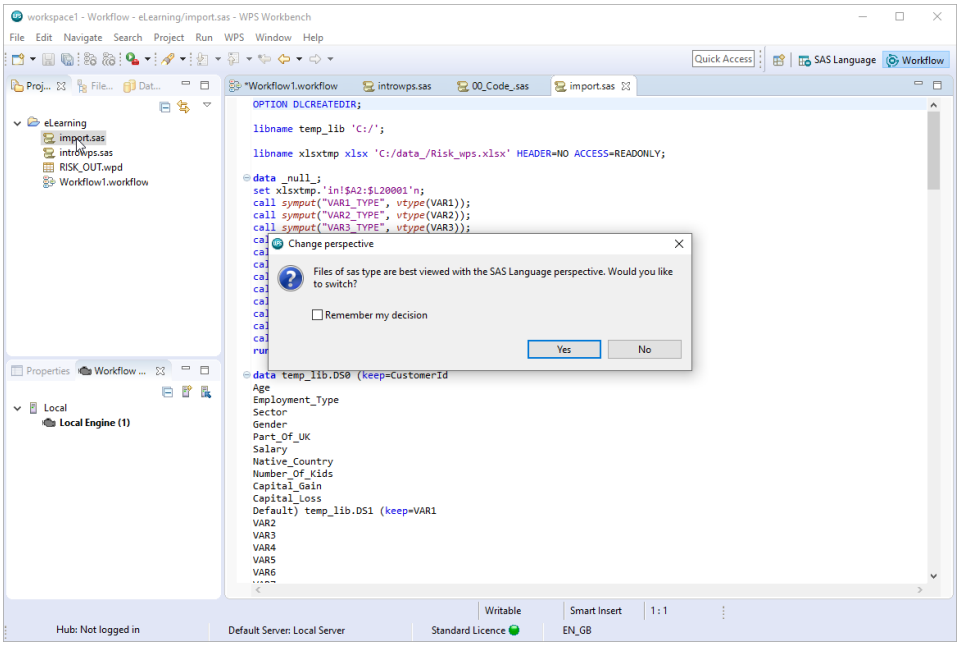

Here the Workflow is exported to a SAS language program to the project folder previously created. The file is named *import_code.*

Figure 43: Export to SAS Language Program




When complete, refreshing the project shows the exported file. This can be easily opened by double clicking.

Figure 44: Opening a code file in the Workflow



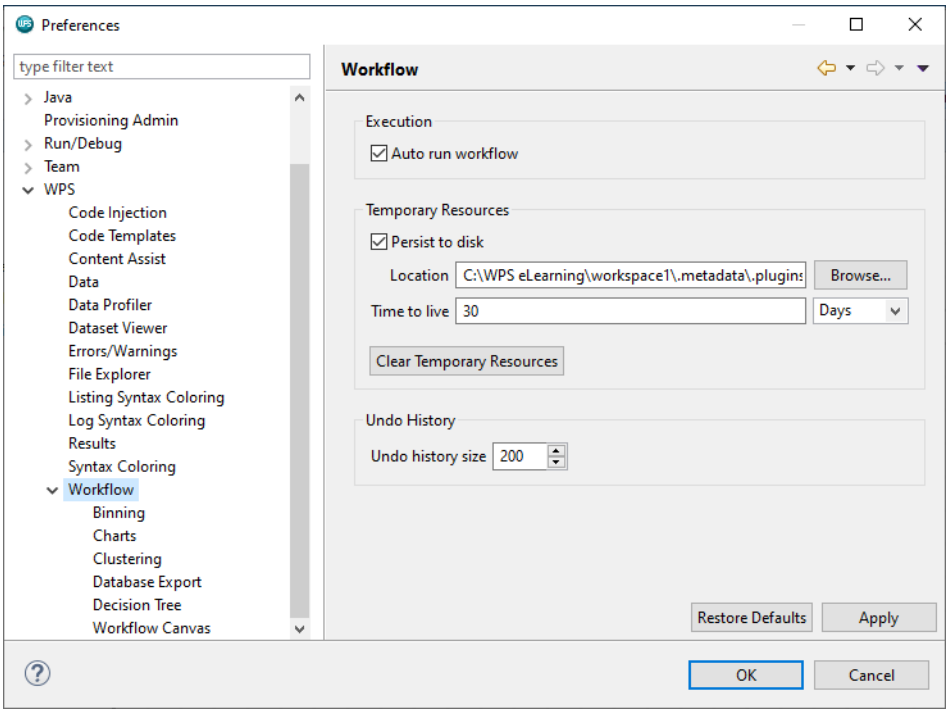

As the file contains code and is being opened in the Workflow a message appears suggesting to open the file in the SAS Language perspective. Clicking the option: Remember my decision will ensure that in future the switch to the SAS Language perspective happens automatically if a code file is opened from the Workflow perspective.

Clicking Yes, opens the file in the SAS Language perspective. A similar message appears if a Workflow is opened from the SAS Language perspective.

Note that if the Workflow is already opened the message will not appear and the perspective will need to be manually selected by clicking the Workflow icon in the top right- hand corner.

Options related specifically to the Workflow can be found from Preferences in the Window menu by expanding ALTAIR and clicking Workflow.

Figure 45: Workflow preferences




Located here are options to auto run the Workflow, which is selected by default. Deselecting Auto run Workflow will mean that each block will need to be manually right clicked and a run option selected.

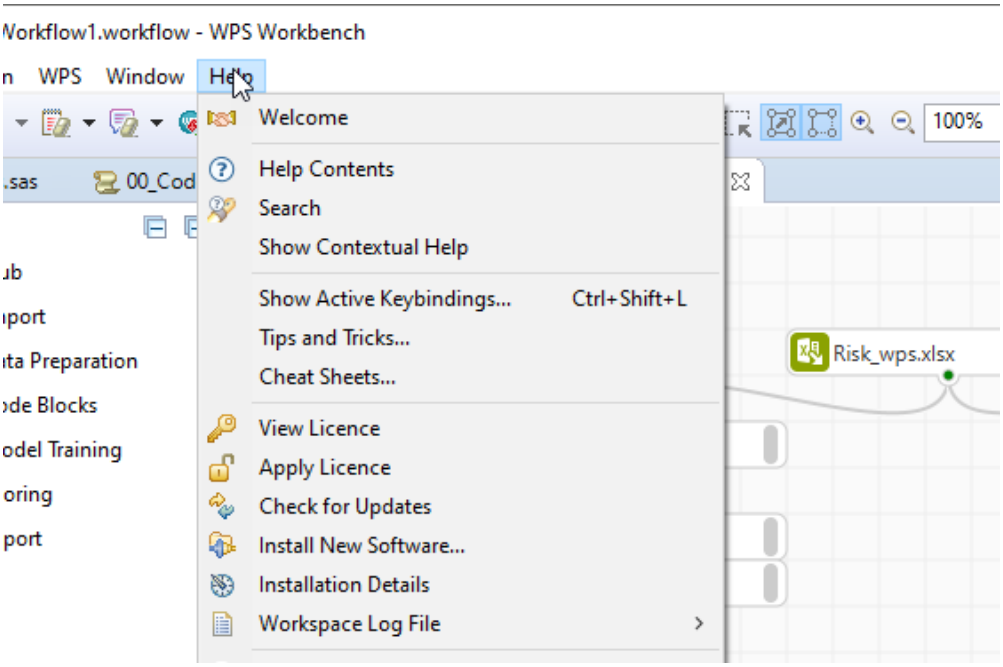
Additionally, options to set the Workflow cache are also evident here. The Workflow is stored in a cache and includes all data and code which means the Workflow can be loaded quickly.

The Workflow cache expires after 30 days but this can be modified. Undo options are also contained here.

Expanding Workflow provides options for Workflow blocks and the Workflow canvas.

The Help menu provides access to help facilities and license information. This menu also includes the option About Altair Analytics Workbench which displays Version information.

Figure 46: Help menu options

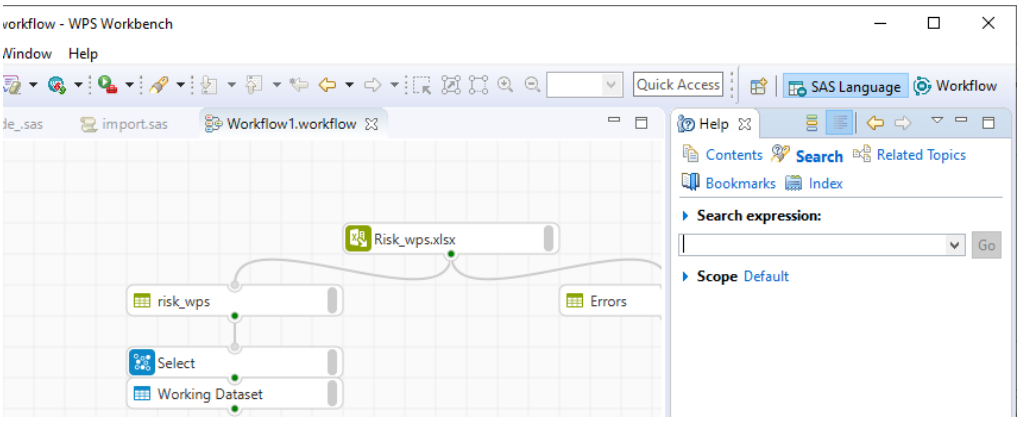



The first Help menu item: Welcome, opens the Welcome page introduced earlier. This is visible when the program is first opened or when switching to a new Workspace.

Help contents opens an expandable set of dictionaries on program use including Whats New. A search bar provides the facility to search for specific content.

Search opens a pane to the right-hand side of the currently open perspective with options to view contents, search, locate related topics and add bookmarks.

Figure 47: Help > Search




The Contents, Search and Related Topics options correspond with options from the Help menu: Help contents, Search and Show contextual help. Show contextual help automatically opens specific topics associated with any item selected in any perspective.

Additional help items include keyboard shortcuts via keyboard bindings, tips and tricks and cheat sheets.

## Summary

This lesson focused on the Altair Analytics Workbench, the SAS Language and Workflow perspectives, common elements, Help and Preferences prior to a demonstration then returning to slides and highlighting points to remember.