



# Overlord3™ User Manual

Software Version 3.x  
Document version 0.6

Tecnoflo.com  
info@tecnoflo.com  
+49 1764 3851 634  
Gartenfelder Str. 29-37 /  
Geb. 39, Workbox 6,  
13599 Berlin, Germany  
www.labsrobots.com

## Table of Contents

1	Getting Started .....	6
1.1	New in this Version .....	6
1.2	Quick Start .....	6
1.3	Using the command palette.....	10
1.4	Saving the procedure .....	12
1.5	Running the procedure .....	14
2	Scheduling.....	15
2.1	Event driven/real-time .....	15
2.2	Static/time driven.....	17
2.3	Dynamic/time driven .....	18
3	Configuration .....	19
3.1	Command Setup tab.....	19
3.1.1	Getting instrument commands .....	19
3.1.2	Adding Instrument Commands .....	21
3.1.3	Removing Commands .....	27
3.1.4	Adding a storage instrument .....	28
3.1.5	Positions.....	37
3.2	Labware Setup .....	40
3.3	Command Setup .....	43
3.4	Options.....	45
3.4.1	Options, General .....	45
3.4.2	Options - Setup .....	49
3.4.3	Options, Schedule.....	51
3.4.4	Options, Email.....	52
3.4.5	Email on Error .....	52
3.4.6	Options, Script .....	54
3.5	Overlord3™ Activation .....	55
3.6	Overlord1 Instrument Driver registration .....	59
4	Commands .....	61
4.1	Decision .....	61
4.2	Loop.....	69
4.2.1	Running a simple loop.....	69
4.2.2	Looping on a condition.....	76
4.3	Call Procedure .....	80

4.4	Exit Procedure .....	83
4.5	Exit Run.....	84
4.6	Initialize Variables.....	84
4.7	Set Variables .....	85
4.8	Message Box .....	90
4.9	User Input .....	95
4.10	Scripts.....	98
4.10.1	IntelliSense.....	100
4.10.2	Using Overlord variables in Scripts.....	101
4.10.3	Code Snippets.....	102
4.11	Timers.....	105
4.11.1	Duration .....	106
4.11.2	Description .....	107
4.11.3	Adjust Timer .....	109
4.11.4	Return Time.....	110
4.11.5	Wait for Timer.....	111
4.12	Loop .....	111
4.13	Move Command .....	111
4.13.1	Setup.....	111
4.13.2	Move Type.....	113
4.13.3	Initialize Procedure .....	114
4.13.4	Change Selected Procedure .....	114
4.13.5	Direct Moves.....	114
4.13.6	Advanced: Enable variable selection on Move Edit dialog.....	116
4.14	Move Command Procedures.....	117
4.14.1	Get Move Command Procedure .....	117
4.14.2	Put Move Command Procedure .....	119
4.15	Adding a Move to a Standard or Schedule Procedure .....	121
4.15.1	Process Number .....	123
4.15.2	Return Barcode .....	123
4.15.3	Use Wait Time .....	125
5	Main Overlord3™ screen .....	127
5.1	Running Overlord3™ .....	127
5.2	File ribbon options .....	128
5.2.1	File options.....	128
5.2.2	Ribbon options.....	129

5.3	Top panel - tab and buttons .....	133
5.3.1	Design tab .....	133
5.3.2	Runtime tab .....	143
5.3.3	Setup tab .....	145
5.3.4	Environmental controls, main additional options .....	145
5.3.5	Environmental controls, help .....	146
5.3.6	Environmental controls, about .....	146
5.4	Right panel .....	147
5.4.1	Variables .....	147
5.4.2	Help .....	151
5.4.3	Devices .....	155
5.5	Runtime information/Error list .....	157
5.6	Command palette .....	158
5.7	Favorites area .....	160
5.8	Palette configuration .....	162
5.9	Overlord3™ window controls .....	164
5.9.1	Environmental controls, thumbtacks .....	164
5.9.2	Work Area controls, Multiple Document Interface .....	165
5.9.3	Tooltips .....	171
5.9.4	Getting and changing properties of a procedure step .....	172
5.9.5	Cut, Copy, Paste, Delete, Properties .....	173
6	Creating Procedures .....	174
6.1	Opening a new procedure .....	174
6.2	Standard Procedure .....	175
6.3	Move Command Procedure .....	175
6.4	Schedule Process Procedure .....	177
6.5	Procedure Flowchart .....	178
6.6	Overlord Variables .....	180
6.7	Actions .....	181
6.8	Example Procedure .....	183
6.9	Advanced Procedure with Incubations and Data Handling .....	189
7	Running Procedures .....	193
7.1	Standard (Event-driven) Procedures .....	193
7.1.1	Run .....	193
7.1.2	Validation .....	195
7.1.3	Error Handling .....	195



7.1.4	Logfile.....	196
7.2	Running Schedule Process (time-driven) Procedures .....	198
8	Frequently Asked Questions.....	199
9	Troubleshooting .....	201
10	External Control .....	202
10.1	Run from Command Line.....	202

## 1 Getting Started

**Overlord3™** is the latest version of the Overlord range of laboratory automation workcell software packages.

Overlord3 is a flowchart-based supervisory software package that allows the control of multiple instruments from one PC in a standard package. Overlord3 utilizes the native software of ancillary equipment for maximum flexibility. Even the most complex system is easy to put together and the most complex instrumentation is simple to integrate.

Overlord3 allows the control of an automated workcell with either:

- event driven, real-time scheduling (as implemented in OVERLORD™ Workstation and Overlord2)
- static, time-resolved scheduling (as implemented in OVERLORD™ Scheduler)
- dynamic, time-resolved scheduling (new option)

### 1.1 New in this Version

#### Features

- Multiple scheduling options
- Treat all sample plates identically
- Parallel processing on multiple instruments
- Advanced run-time error checking and recovery
- Empowering advanced users

#### Benefits

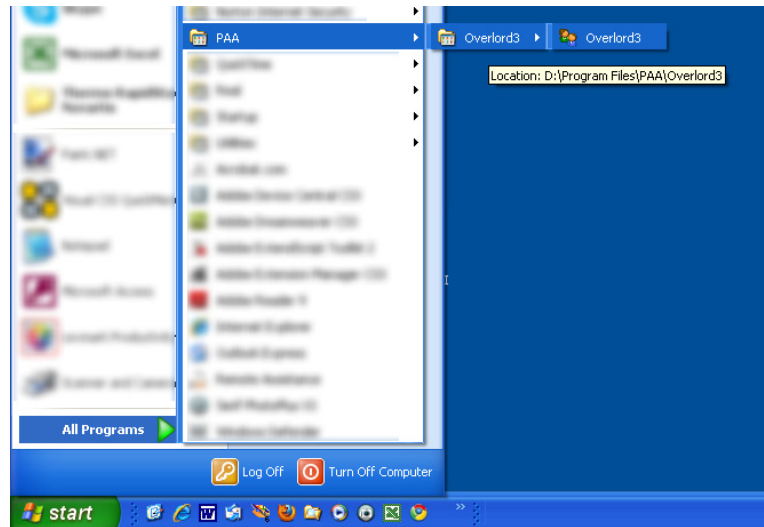
- Ease of programming
- Rapid assay development
- Increased plate throughput
- Fully recoverable from device errors
- Powerful design environment
- Support for device pooling

### 1.2 Quick Start

It is really easy to create an Overlord3 procedure. Follow these steps to familiarize yourself with the basics.

Start Overlord3.

There should be an entry on the Start Menu under START → All Programs → PAA → Overlord3



**Figure 1.1 - Start menu**

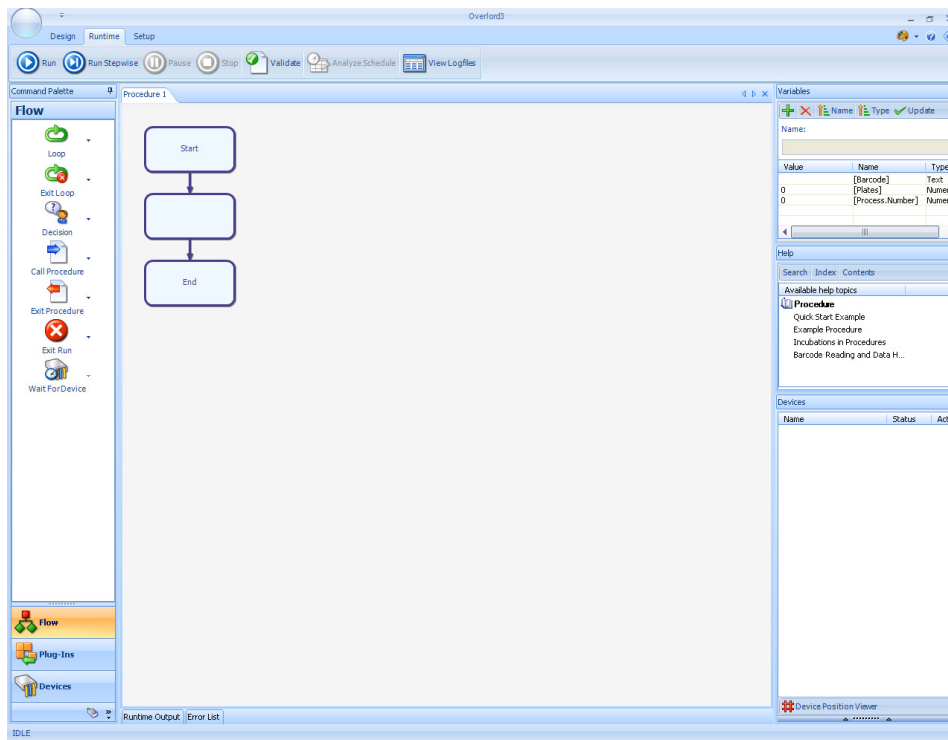
Upon starting you will see the splash screen (Figure 1.2 - Splash screen).



**Figure 1.2 - Splash screen**

The splash screen contains the detail of the loading sequence (yellow) and the Overlord3 version number (red).

Then the Overlord3 main screen should appear (Figure 1.3 - Overlord Main Screen).



**Figure 1.3 - Overlord Main Screen**

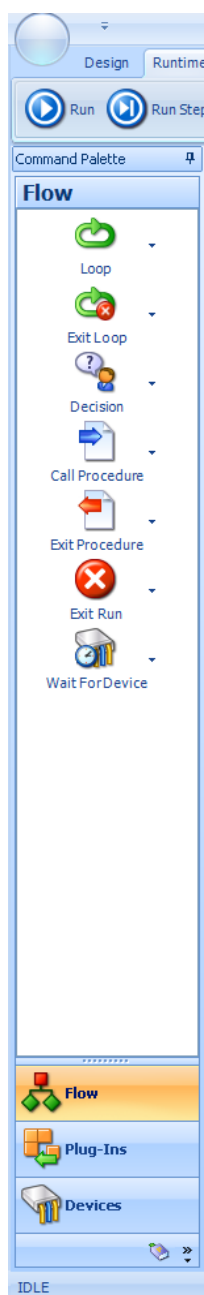
On the left hand-side of the window is the **Command Palette** and here you will find the Overlord3 Commands, which allow you to build a procedure.

These commands are listed into three different categories, **Flow**, **Plug-ins** and **Devices**.

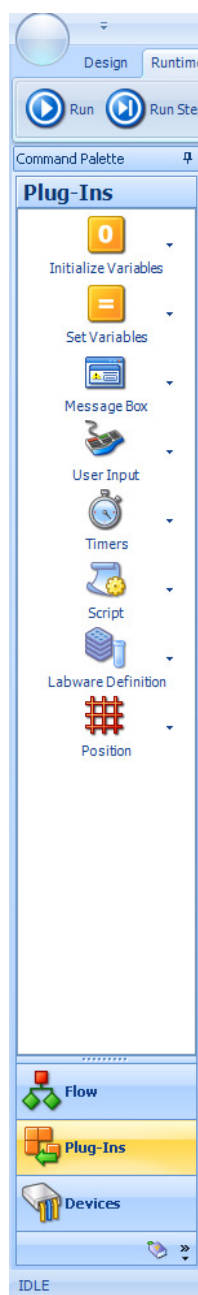
The start default is Flow (Figure 1.4 - Flow palette).

Clicking on the **Plug-Ins** icon on the Command Palette will show the available Plug-In commands (Figure 1.5 - Plug-Ins palette).

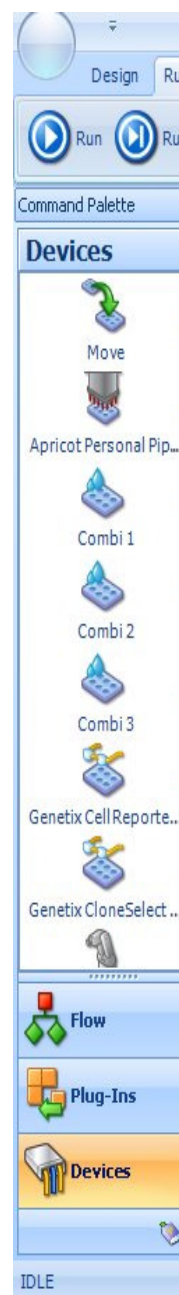
If you wish to see and use the installed instruments, select the **Devices** palette (Figure 1.6 - Devices palette).



**Figure 1.4 - Flow palette**



**Figure 1.5 - Plug-Ins palette**



**Figure 1.6 - Devices palette**

### 1.3 Using the command palette

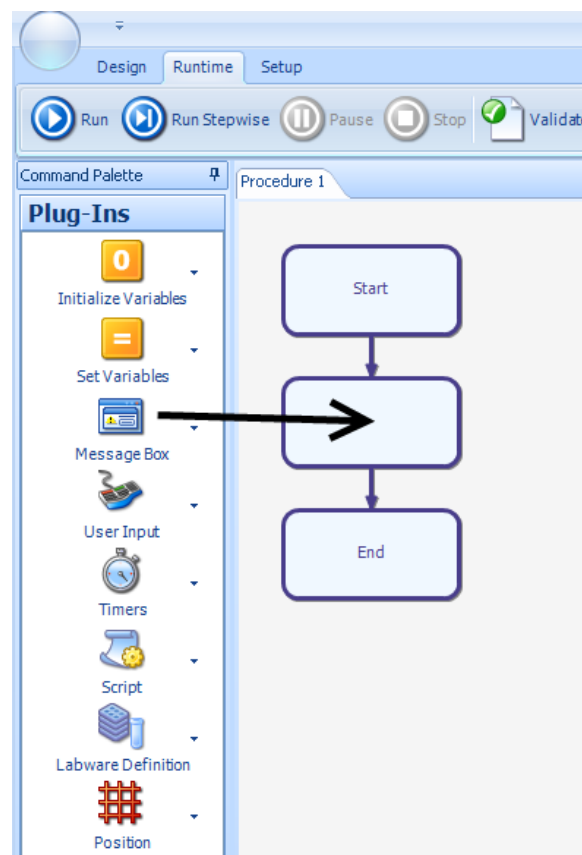
For example the Message Box command (show a message to the user at runtime).

From the **Plug-Ins** palette, drag the **Message Box** command across onto the empty **Action** box on the **Procedure** Flowchart.



**Figure 1.7 - Message Box icon**

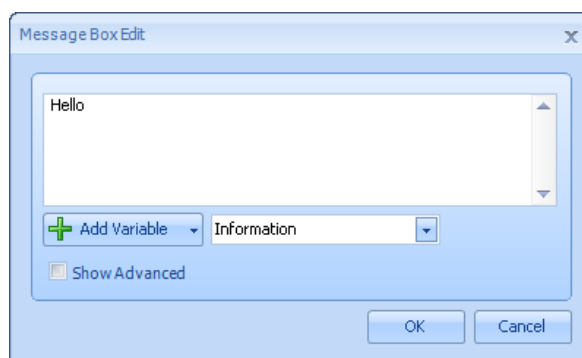
This is achieved by putting the mouse cursor on the **Message Box** icon, hold the left hand mouse down, move the cursor over the required **Action** box in the Overlord3 **Procedure** work area, and release the mouse button. The move command is represented by the plus icon.



**Figure 1.8 - using the message box command**

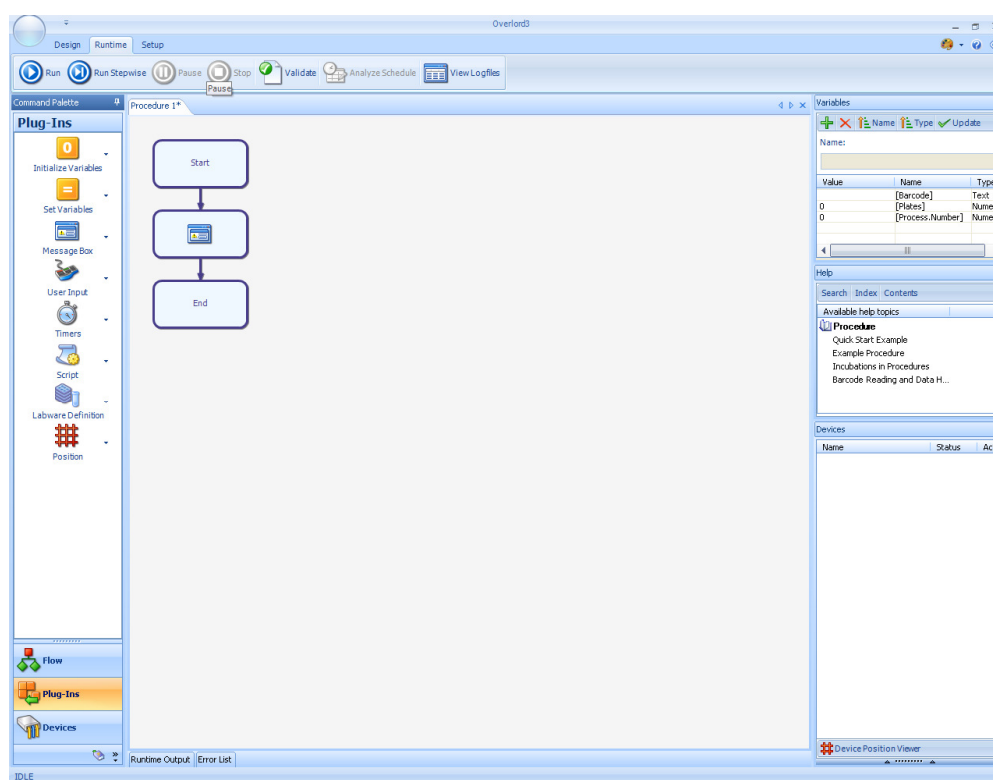
The Edit properties for this command will then appear.

Type in the Empty white text box **"Hello"** and then click on the OK button (Figure 1.9 - Message Box Edit).



**Figure 1.9 - Message Box Edit**

The window will then close and the procedure flowchart should have the first Action in the Overlord3 Procedure (Figure 1.10 - Simple Procedure).



**Figure 1.10 - Simple Procedure**

## 1.4 Saving the procedure

Click on the File **Function Disc** in the top left hand corner of the window (Figure 1.11 - Save As function location).

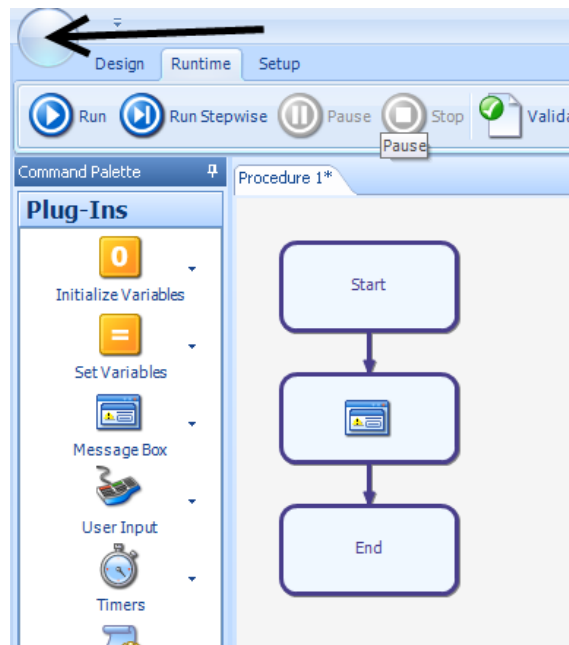


Figure 1.11 - Save As function location

In the drop down menu select **Save As** (Figure 1.12 - Save As menu).

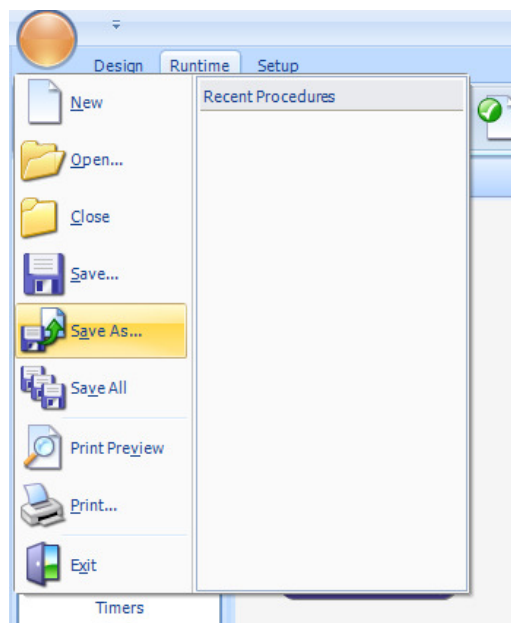
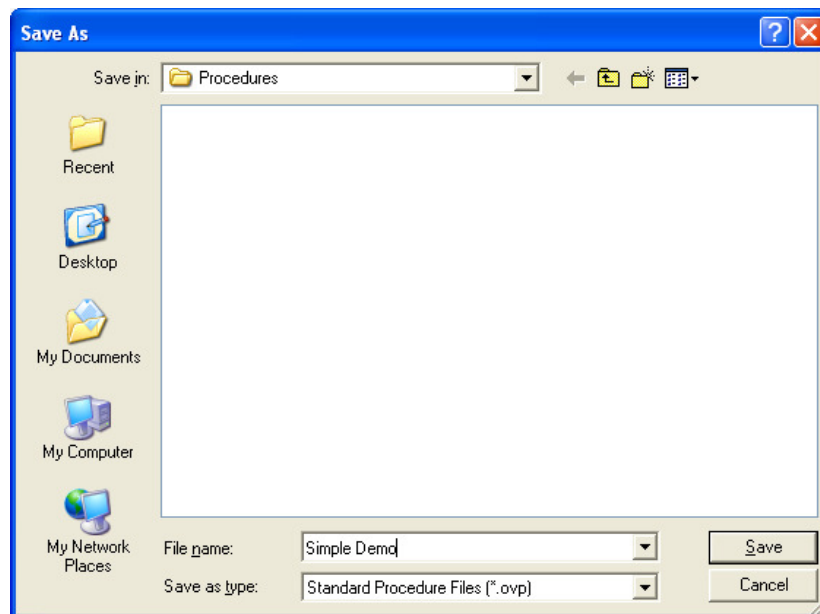


Figure 1.12 - Save As menu

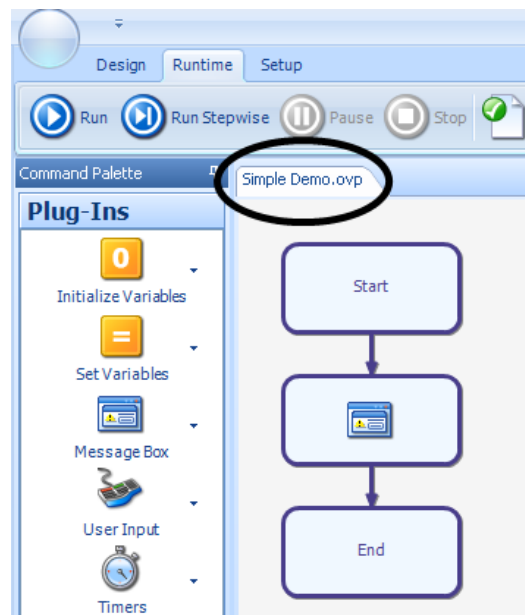


The Save As dialog window will appear (Figure 1.13 - Save As dialog window).



**Figure 1.13 - Save As dialog window**

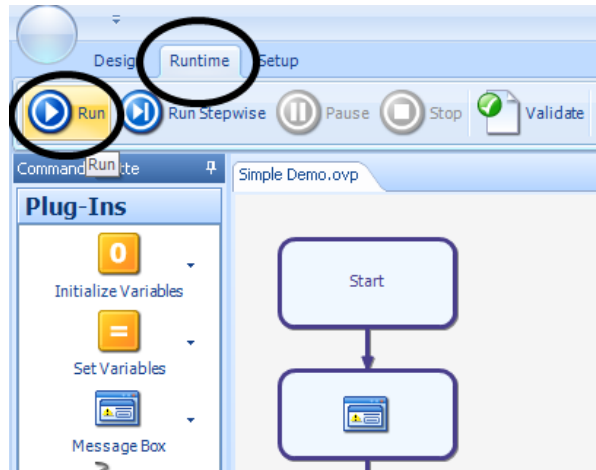
Enter the procedure name 'Simple Demo' and select **Save**. The new procedure name will then appear on the name tab (Figure 1.14 - procedure name tab).



**Figure 1.14 - procedure name tab**

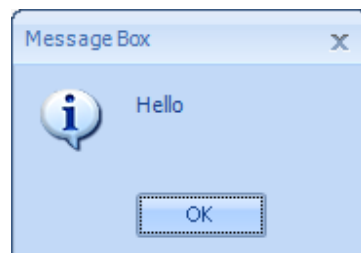
## 1.5 Running the procedure

Select the tab '**Runtime**' from the toolbar and click on the **Run** button (Figure 1.15 - Running a procedure).



**Figure 1.15 - Running a procedure**

A message box saying "**Hello**" should appear. Click Ok on this message box and the run will complete.



**Figure 1.16 - Running a Procedure**

Add another Message Box now between the existing Message box Action and the End Action.

When you drag the Message Box across from the Command Palette, the place where you want to insert will be highlighted red.

The Message box Edit dialog will appear again.

This time enter "**Hello again**" and then click OK. Save this procedure and click Run. "**Hello**" should appear and then clicking OK will bring up the next dialog box saying "**Hello again**". Click OK and the run will complete.

If you have a security key error, contact PAA for support.

## 2 Scheduling

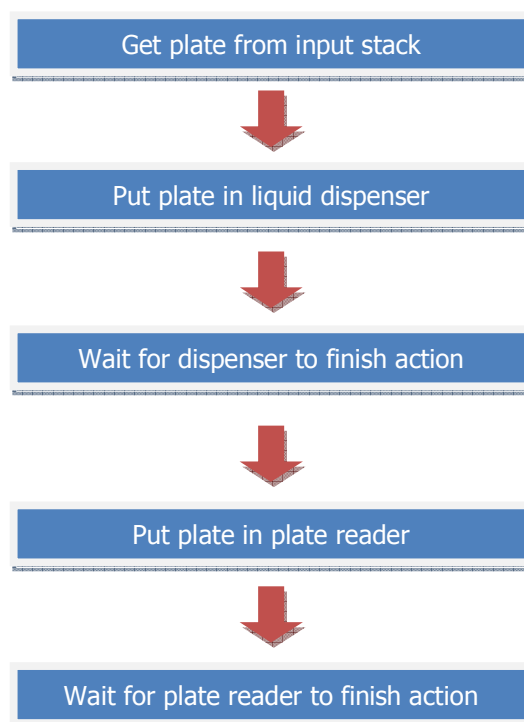
Scheduling of laboratory equipment into a fully integrated automated system required integration software. There are three major types:

- Event driven/real-time
- Static/time driven
- Dynamic/time driven

### 2.1 Event driven/real-time

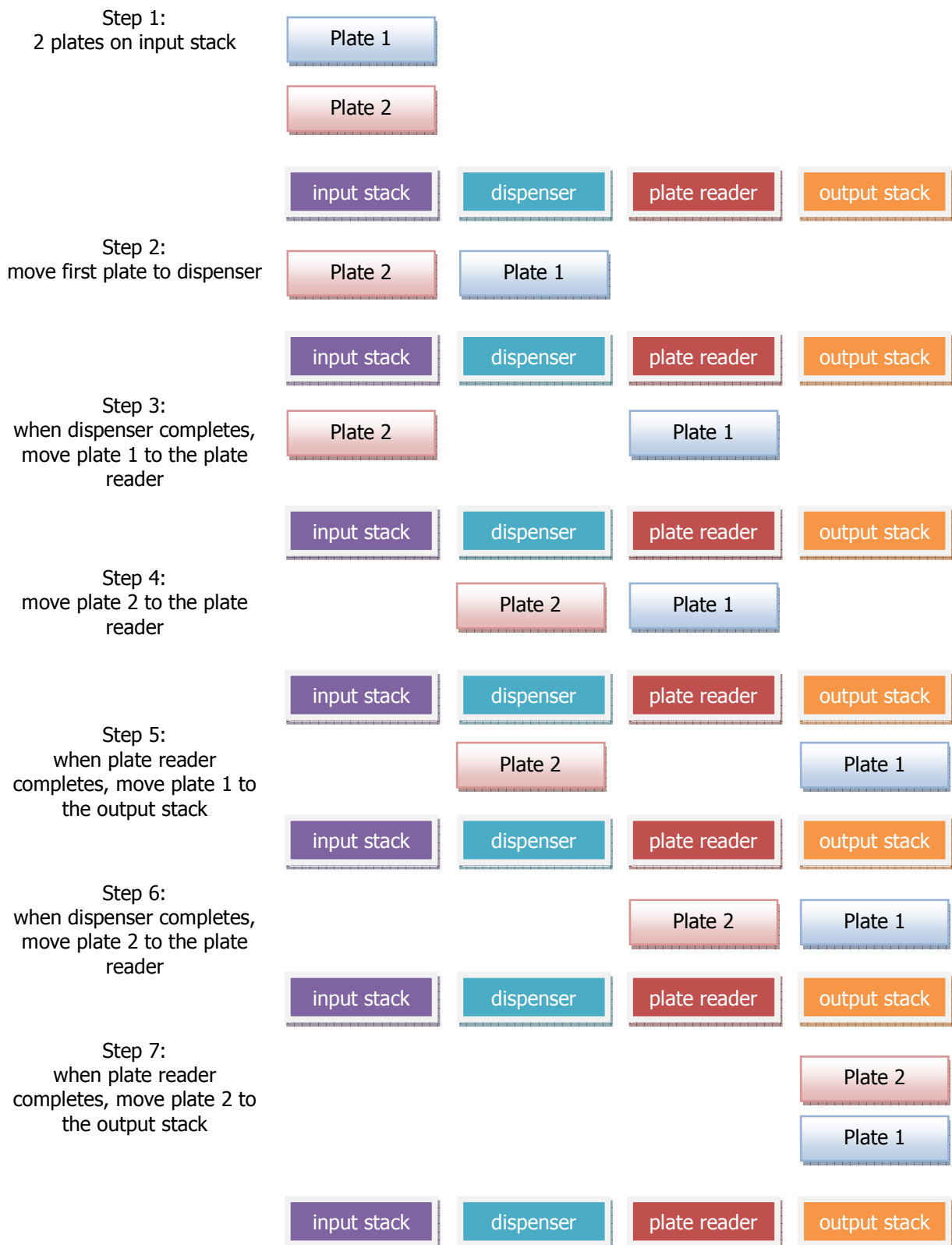
This type of process is linear. The instructions are processed sequentially, and are best represented by a flow chart. In its simplest form, the next step of the process is not started until the last step is completed. However, this would be a very slow way of completing a complex process with multiple steps. For example, if there is incubation for 1 hour, nothing would happen until that 1 hour waiting time has elapsed. So the process is controlled by a series of events, so the next process does not start until event has been completed.

In a simple example, if we wish to process a stack of plates through a dispenser and then a plate reader, the linear process would be ...



There would be wasted time in this process because the plate reader is unused during the dispensing step, and the dispenser is unused during the plate reading step.

As this process is a two stage process, then as the first plate is moved from the dispenser to the plate reader, another plate can be moved from the input stack to the dispenser.



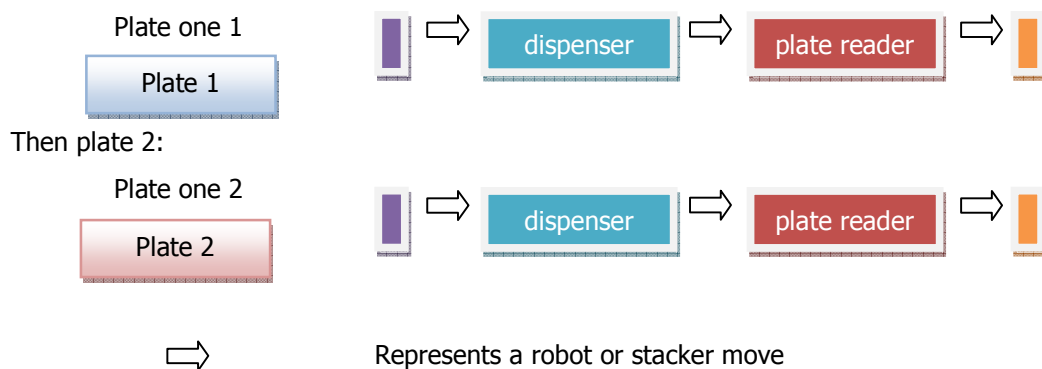
The process is then run as a loop until all the plates are processed. This technique, known as round robin, allows the plates to be processed quickly, but does NOT allow any timings to be taken into account.

These types of schedulers are simple algorithms and are very rapid in completing a process. No time is wasted because as soon as one process is completed, the next process starts. These schedulers are very flexible in what process can be modelled, but can get complex to set up because of the flexibility available. Processes that have more than one critical time can be difficult, or impossible, to encode in this type of scheduler.

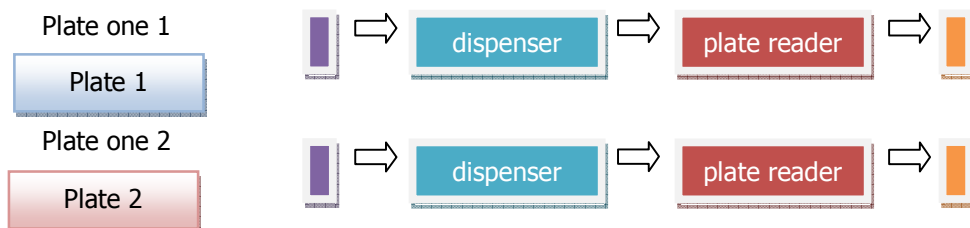
To summarise, when the user presses the go button, the process is defined but the timing isn't. This scheduler is process centric.

## 2.2 Static/time driven

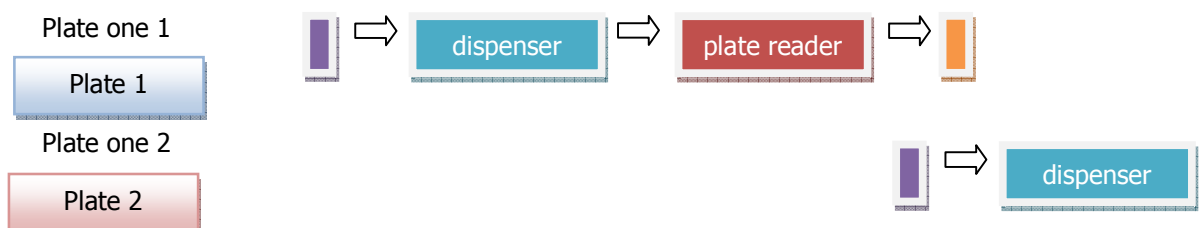
A static scheduler pre-processes the run and lays out the robot and instrument moves before the process starts. This type of scheduler ensures that all the resources are free to be used when required during the process. Taking the example above, plate 1 would be processed:



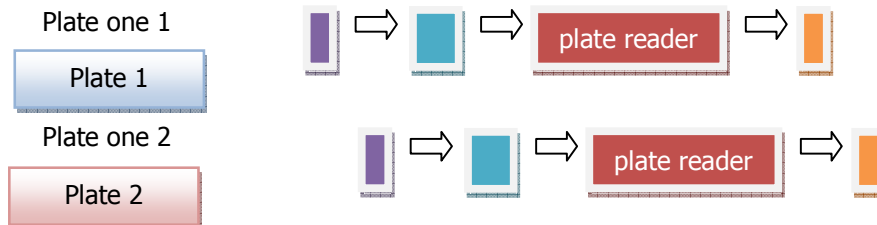
However when put together



There would be a clash of resources, so the process has to be scheduler to make sure the resources are available:



The offset is because the time requirements are the same, the robot can't start work on the second plate until the first plate is finished. Often the time requirements of the instruments and scheduling becomes more efficient.



For maximum accuracy, all the moves and instrument processing times have to be measured. The calculation of the processing time requires a sophisticated algorithm.

To summarise, when the user presses the go button, the process and timings are defined. This scheduler is sample centric.

## 2.3 *Dynamic/time driven*

Dynamic schedulers try to combine the best of both event-driven and static schedulers by starting as a static scheduled process and then allowing rescheduling on the fly. These schedulers are very hungry for computer resources during the run.

With a pre-emptive scheduler, the system runs to a planned time schedule, with total control of the time windows for each element in the system. There is resource sharing and there will never be any conflict in the use in the system resource (unless there is an error, when under certain circumstances the time windows will shift). Uses include; cell culture and processing.

The strength of this type of schedule is the guarantee of samples being treated in an identical way. There is little scope for decisions in the schedule. A pre-emptive is TIME driven.

## 3 Configuration

This section describes the configuration commands found in the:

- ▶ **Command Setup** tab.
- ▶ **Labware Setup** tab
- ▶ **Options** tab



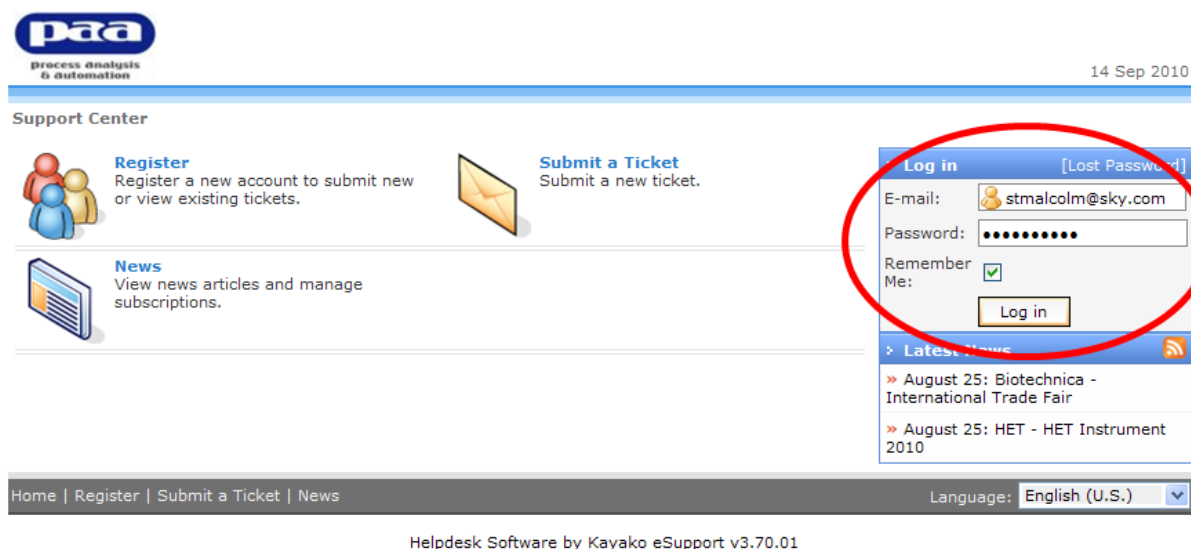
### 3.1 Command Setup tab

This tab option allows the user to configure the attached instruments.

#### 3.1.1 Getting instrument commands

There is a library of available instrument commands available to users for 1 year after purchase of **Overlord3™** and for current service contract holders. The library is held at **paa.helpserve.com**, and user access will be on product purchase.

Goto **paa.helpserve.com** and login (**Error! Reference source not found.**):



Once you have logged in successfully, you will see the options screen. Enter the instrument driver that you require into the **Search** box.



14 Sep 2010

## Support Center

Logged in successfully

**View Tickets**  
Submit new tickets, view existing tickets or create new replies.

**Submit a Ticket**  
Submit a new ticket.

**Knowledgebase**  
Search support articles and find answers to frequently asked questions.

**News**  
View news articles and manage subscriptions.

**Downloads**  
View our library of file downloads and links.

Latest Download Files	Date Added
VT Scanner - Scan Control Exe Config File	08 Sep 2010 04:53 PM
VT Scanner - Scan Control Exe	08 Sep 2010 04:52 PM
VT Scanner Command	08 Sep 2010 04:41 PM
EnVision Command	26 Aug 2010 09:47 AM

Popular Knowledgebase Articles	Views
SQL Server error message appears when trying to start Overlord2	67
Overlord2/SQL Installation Issue	51
SQL Error "Failed to Generate user instance of SQL Server."	43
Installation of SCARA KINEDx Robot Control Software	23

Latest Knowledgebase Articles	Date Added
-------------------------------	------------

**My Account** [Logout]

Logged In: **Malcolm Crook**

My Account

Change Password

**Search**

Thermo Multidrop

-- Entire Support Site --

**Latest News**

» August 25: Biotechnica - International Trade Fair

» August 25: HET - HET Instrument 2010

The selection of this search will bring up all information relevant to the selected instrument, in this case the Thermo Multidrop (paa.helpserve.com entry for Thermo Multidrop):






14 Sep 2010

## Support Center


> Search Query: Thermo Multidrop

Time Taken: 0.0065 Seconds

### Knowledgebase

-  Timeout occurred while waiting for action to complete: Multidrop Combi Error Code 3  
SymptomsGet error message "Timeout occurred while waiting for action to complete: Multidrop Combi Error Code 3. Please refer to the Combi User Guide." when trying to run a dispense on the Thermo Multidrop Combi.SolutionThe code 3 error indicates


### Downloads


-  Multidrop  
Overlord Driver ovThermoLSMultidrop.dll 3.1.0.4 Used for: Multidrop Micro, Multidrop Deep Well, Multidrop 384
-  Multidrop Help  
Overlord Driver Help ThermoMultidrop.help.zip Used for: Multidrop Micro, Multidrop Deep Well, Multidrop 384
-  Multidrop EXE  
Required for Overlord Driver ThermoLSMultidrop.exe 3.0.0.14 Used for: Multidrop Micro, Multidrop Deep Well, Multidrop 384
-  Multidrop Combi (Overlord1)  
Overlord Driver ovThermoLSMultidropCombi.dll 1.0.0.3
-  Multidrop Combi (Overlord2)  
Overlord2 Command Overlord.Commands.ThermoMultidropCombi.dll 1.2.0.16
-  Multidrop Combi nL  
Overlord2 Command Overlord.Commands.ThermoMultidropCombiNL.dll 1.0.0.4

 Back

> My Account [Logout]

Logged In: **Malcolm Crook**

 My Account

 Change Password

> Search

-- Entire Support Site --

[Home](#) | [View Tickets](#) | [Submit a Ticket](#) | [Knowledgebase](#) | [News](#) | [Downloads](#)

Language: English (U.S.)

Helpdesk Software by Kayako eSupport v3.70.01

## paa.helpserve.com entry for Thermo Multidrop

From here you are able to download any of the available drivers, help files and support files to run any of the extensive Overlord instrument driver library.

Where possible ALWAYS use the Overlord2 (.NET) driver version.

When downloading place:

Overlord1 drivers in **c:\overlord\drivers**

Overlord1 help files in **c:\overlord\help**

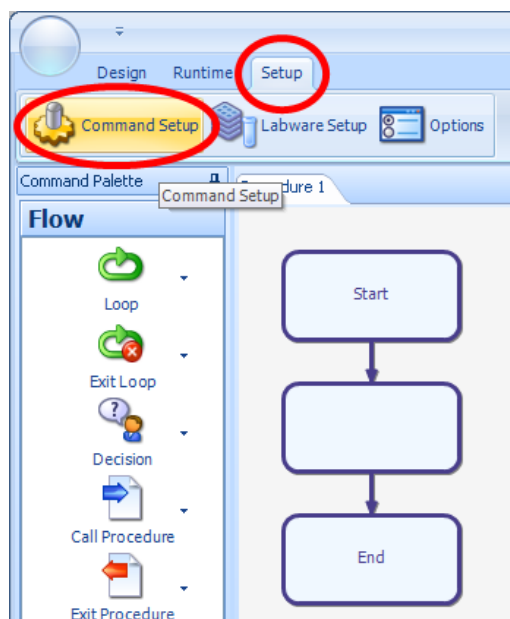
Overlord2 drivers in **c:\program files\paa\overlord3\Commands**

Overlord2 help files in **c:\program files\paa\overlord3\help**

With help files, once this file is installed it will be available from the Help pane on the main screen once the command has been selected in the Command Palette.

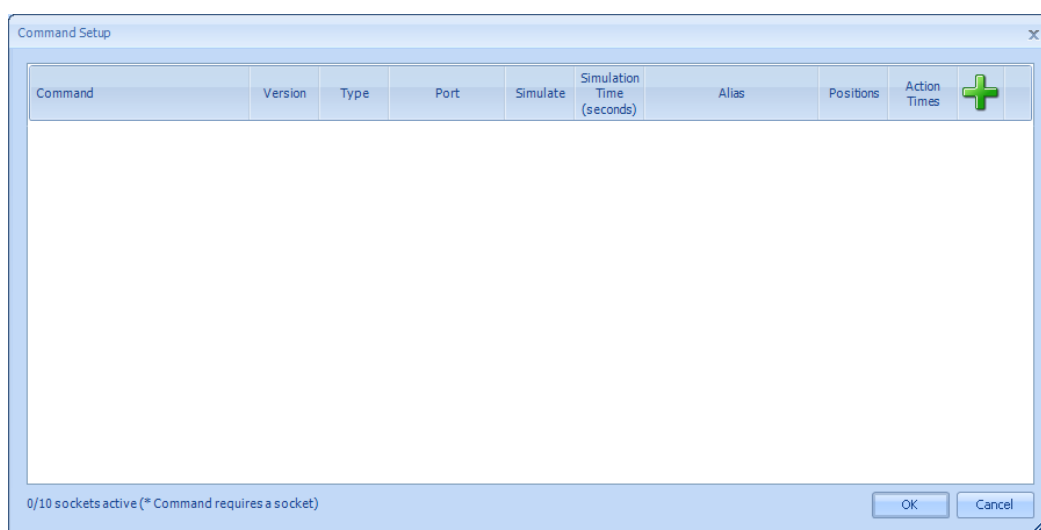
## 3.1.2 Adding Instrument Commands

To add a new command (instrument or data plug-in) into **Overlord3**, click on the **Setup** tab and then the **Command Setup** button (Command Setup Options button):



**Command Setup Options button**

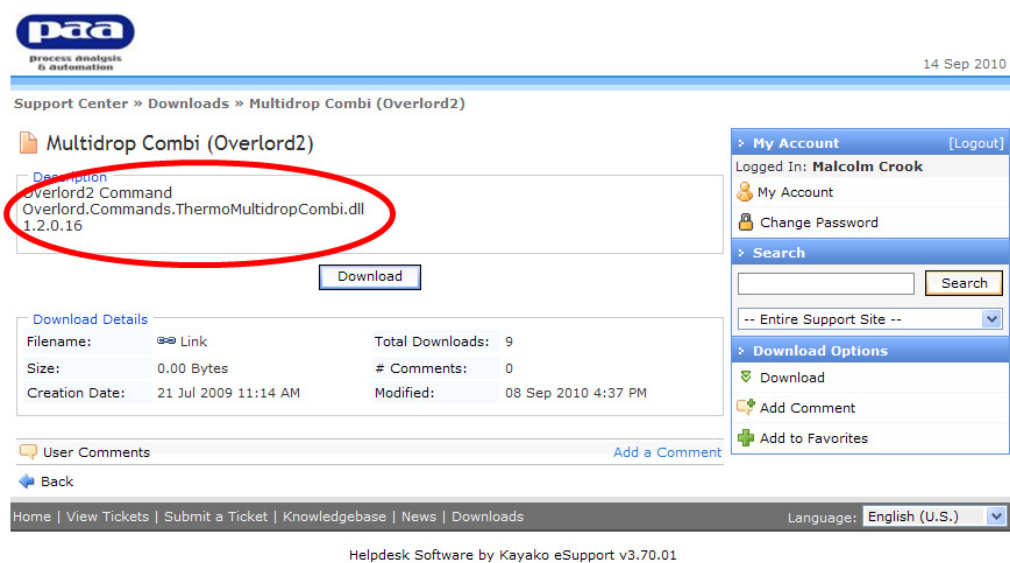

This will bring up the **Command Setup** dialog box which will display **Commands** that have already been selected. If this is a fresh installation, the display will be empty (Commands Setup dialog box):



**Commands Setup dialog box**

The text box in the bottom left informs the user how many sockets are used and how many instrument sockets are available for use in the current license.

The dialog shows a number of properties relating to the instrument command.

Command	Instrument driver name
<b>Version</b>	<p>Software version number (use to check you have the latest version) The version number is included in the paa.helpserve.com download screen</p>  <p>for downloading new drivers, please see download section (Section 3.1.1 Getting instrument commands).</p>
<b>Type</b>	Communication type, Serial, TCP/IP. The port value differs depending upon the communication type of the command. For instance, a command that uses TCP/IP would have the IP address entered in as the port value. Refer to the Readme.txt for each command that is found on the PAA website.
<b>Port</b>	Connection com port 1-n
<b>Simulate</b>	Allows the system to be tested
<b>Simulate Time (secs)</b>	Estimated time for an action when Overlord3™ is being used in time resolved mode.
<b>Alias</b>	The name used in Overlord3™ can be changed. This can be used to reduce the length of the instrument driver name or identify multiple instruments of the same type. The Alias name for each command by default is setup with the name of the Command. This can be changed though and is typically changed if there is more than one of this command on the system. For example if there was two Biotek Washers, the first command would have the Alias "Washer 1"; the second Washer would have the name "Washer 2".
<b>Positions</b>	Allow the supervisor to setup the number of positions on the instrument.
<b>Action Time</b>	Every time an Action is run in Overlord3™ the time duration (in seconds) is recorded. This information is used when running Schedule Procedures, to calculate the schedule for the run.
	Add a new instrument in the <b>Command Setup</b> dialog box.

To add a command, click on the green cross in the top right corner of this window. Overlord3 will then scan for Overlord Drivers in two locations.

Command folder in the folder the Overlord.Main.exe resides in – (in ***X:*** ***Program Files|PAA|Overlord3|Commands***, where ***X*** is the installation directory, usually ***C***)

Overlord1 Driver folder (in ***X:*** ***Overlord|Drivers*** where ***X*** is the installation directory, usually ***C***)

Overlord3 can use **Overlord2** Commands, which have been programmed with the .NET Framework. These are known as .NET native commands and will have following name.

***Overlord.Command.<DeviceName>.dll***

***<DeviceName>*** is the name of the device or instrument this command will control. For example, the driver to control the Deerac Latitude is called ***Overlord.Command.DeeracLatitude.dll***. If this file is found in the Commands folder, normally located at C:\Program Files\PAA\Overlord3\Commands, it will be available to be used in Overlord3.

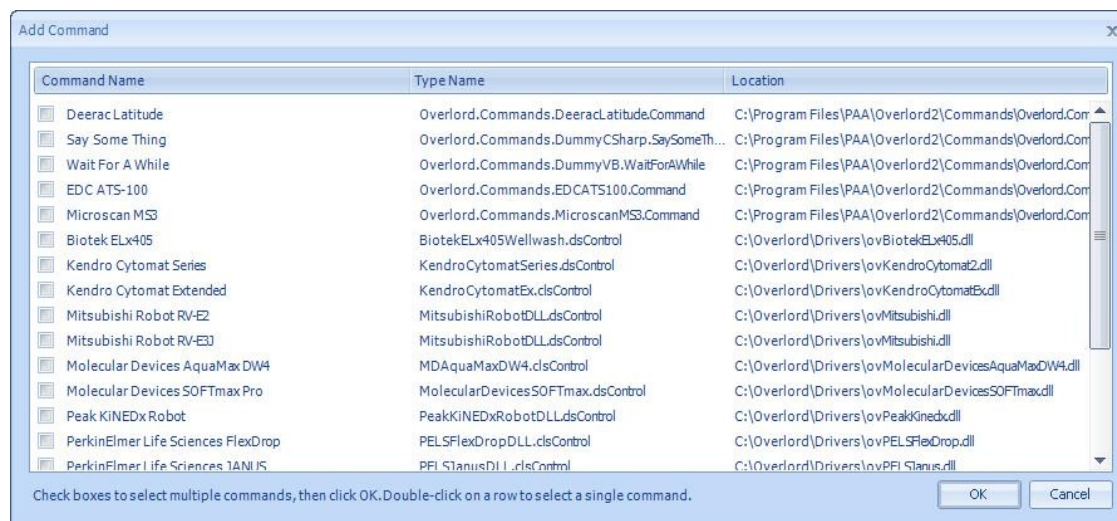
**Overlord3** can also use **Overlord1** instrument driver. These drivers are known as COM Drivers and will have the following name.

***ov<DeviceName>.dll***

If this file is found in the Drivers folder in ***C:*** ***Overlord|Drivers***, it will be available to use in **Overlord3** provided it has been registered. COM Drivers need to be registered on the PC they are being used on. This can be completed in two ways, both are explained in a later section (Section 3.6 Overlord1 Instrument Driver registration).

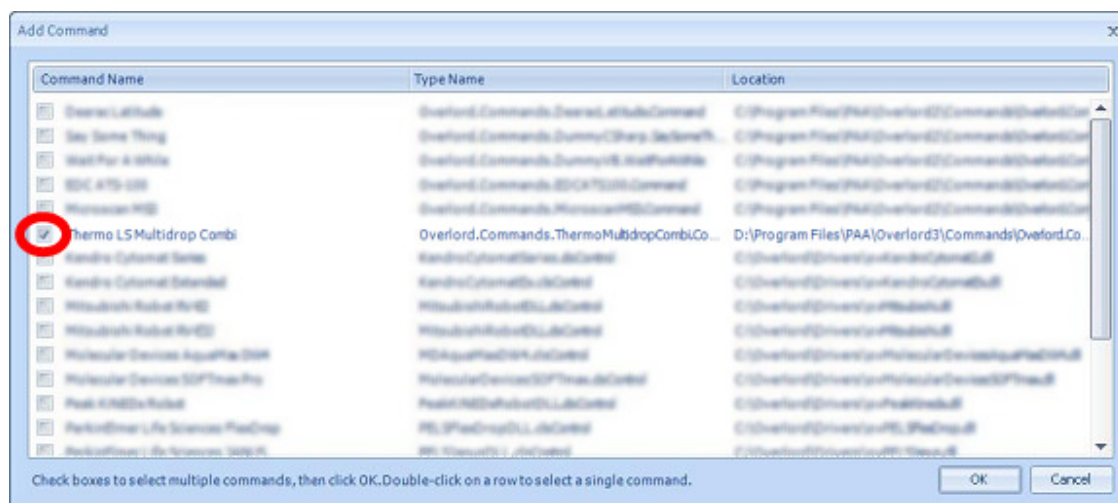
In addition, it is possible that there will be customer specific code modules (**Plug-Ins Commands**) in the command list. When these are selected, the command will be installed in the Plug-Ins palette, not the Device palette. These Plug-Ins modules are for specific customer requirements and are not installed on all systems.

Once **Overlord3** has finished scanning these locations, the **Add Commands** dialog box will appear listing all the Commands that have been found (Add Command):



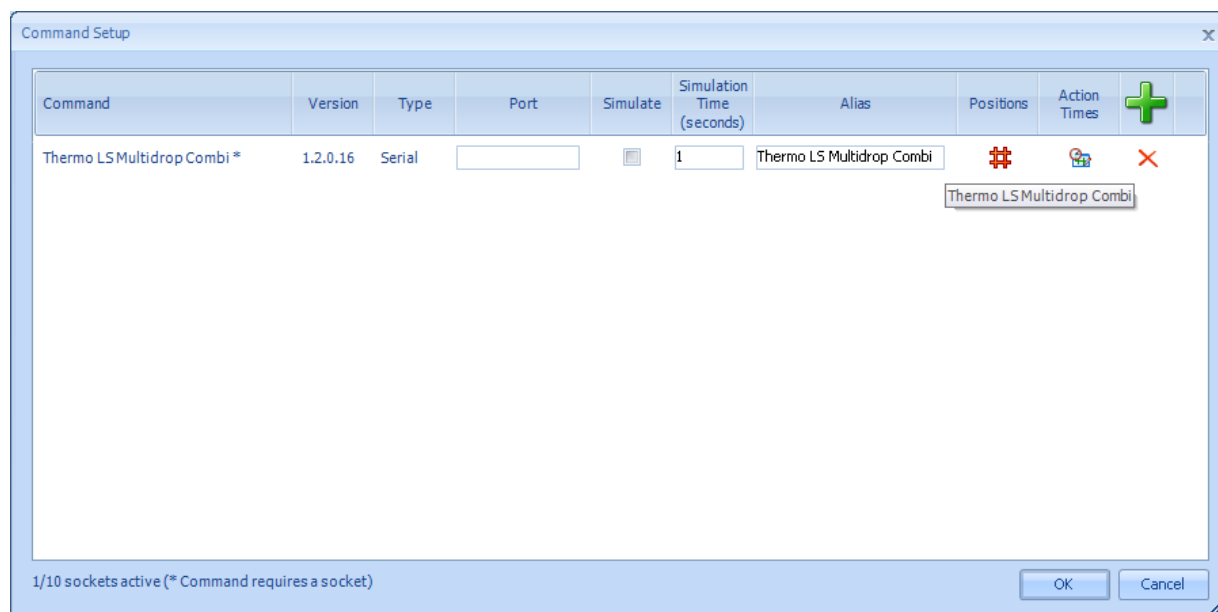
**Add Command**

Select the Command you require by ticking the check box to the left of the **Command Name** and then click **OK** (Select the required command):

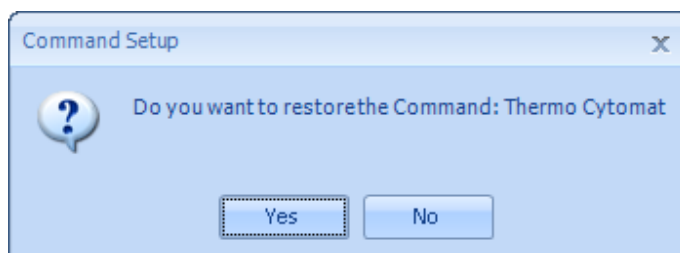


**Select the required command**

This command will then appear in the Command Setup Dialog. In the example below, the Thermo Multidrop has been added to the instrument list (**Error! Reference source not found.**):

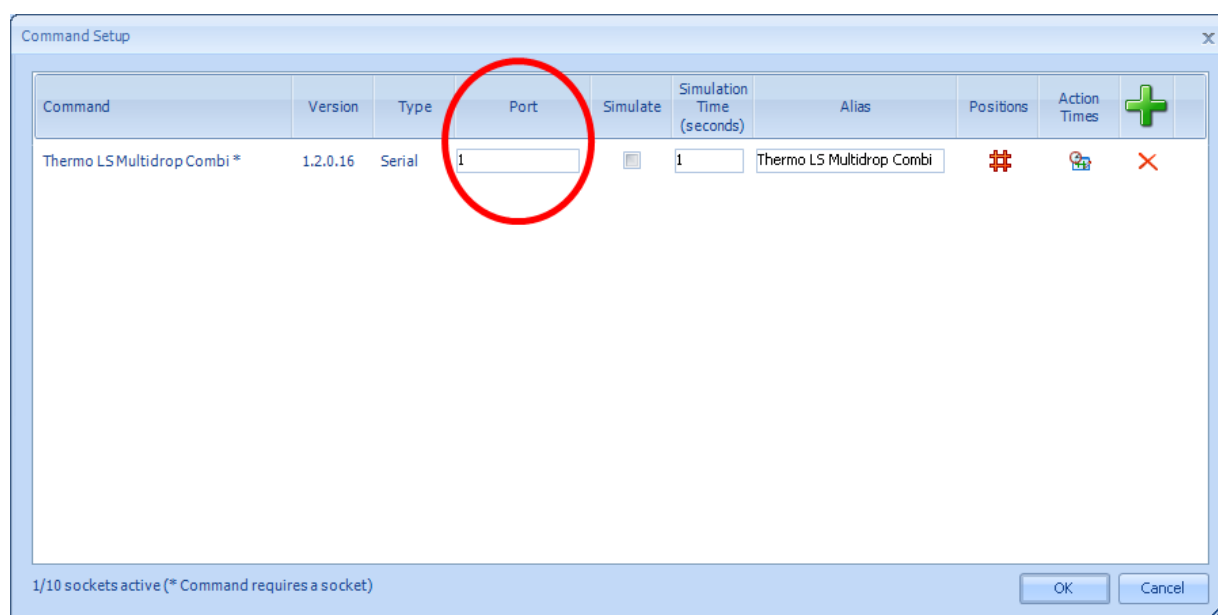


If the instrument is added to the system, but has been added and then removed in the past, the user will be offered to restore the command. If the command has been used in **Overlord3™** procedures, choose Yes and this will allow the previous procedures to function (Restore the previous installation of the instrument):



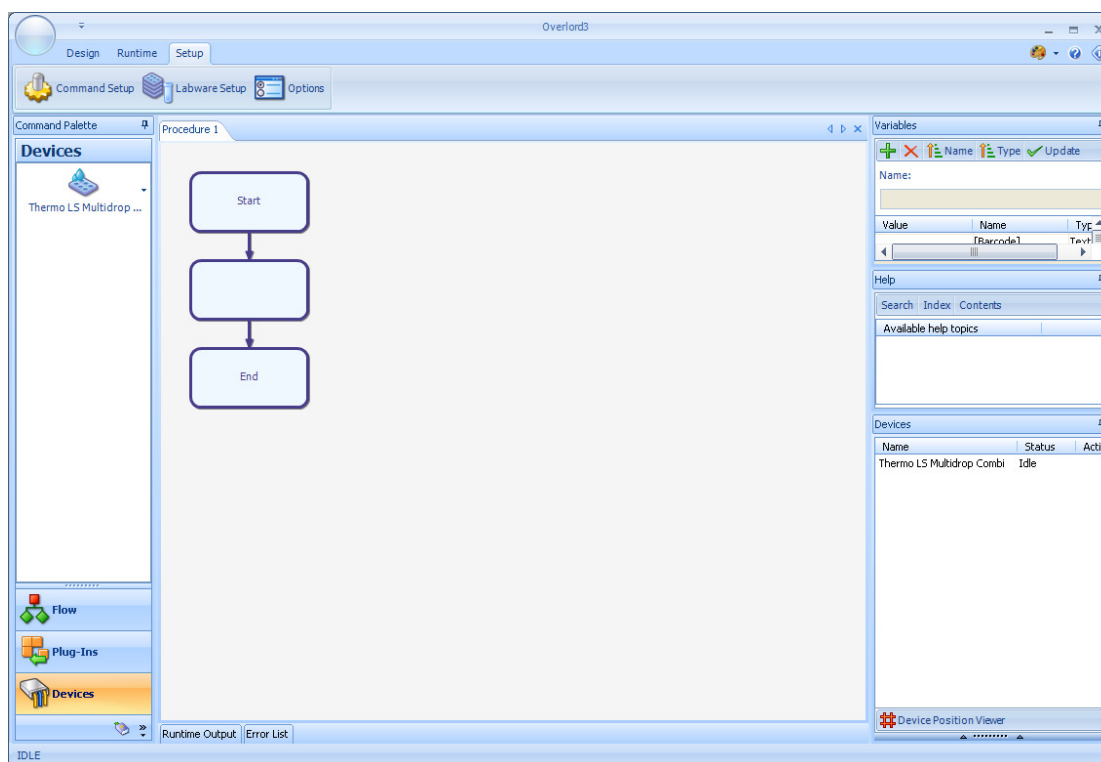
**Restore the previous installation of the instrument**

As the Thermo Multidrop washer uses serial communication, the port of the PC needs to be entered into the Port text box. Enter the port number by clicking the cursor in the Port box and enter the physical connection port number (Enter port number in Command Setup):



**Enter port number in Command Setup**

After adding all of the commands, click OK on the Command Setup dialog to save these settings. The new commands will now be available to add into a procedure. Depending upon the type of command, they will either be in **Plug-Ins** or **Devices** section of the Command Palette (Thermo Multidrop instrument now available to use):



**Thermo Multidrop instrument now available to use**

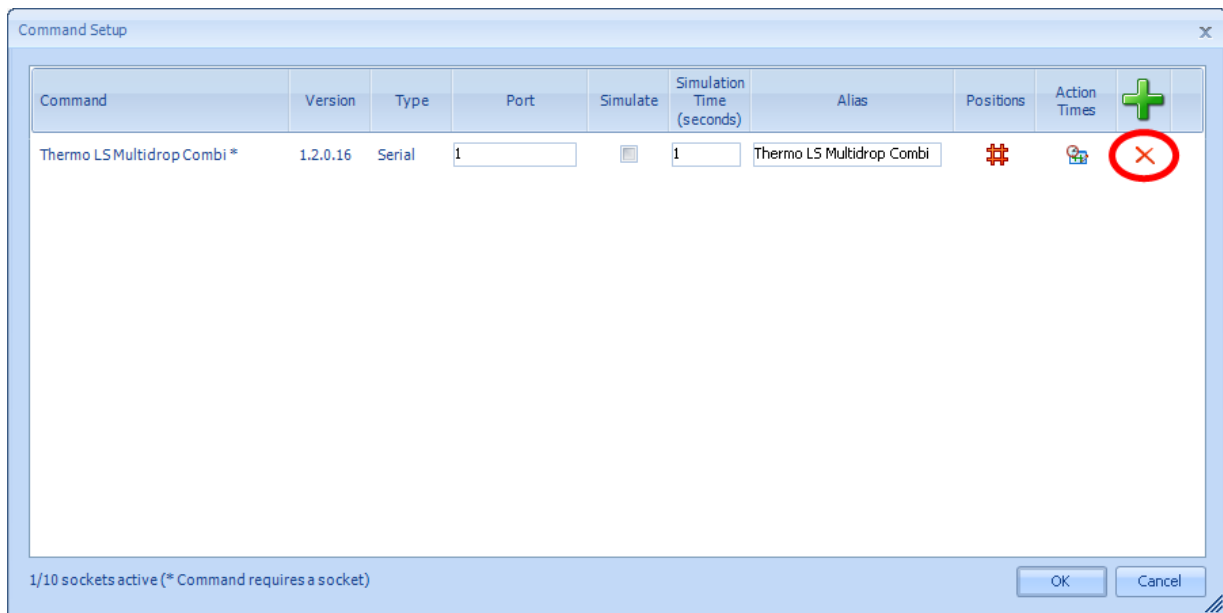
## 3.1.3 Removing Commands

The Overlord3 licence allows the user to run a number of instruments on the system, without stating which instruments are being used. Sometimes the user may require to change the instrument command driver.

To remove an installed driver, select the red cross at the far right of the instrument description line (Removing a Command for the configuration). The instrument will be removed from the configuration.

To add a new instrument, follow the instructions in the section above (Section 3.1.2, Adding Instrument Commands).

**DO NOT REMOVE THE DRIVER FILENAME FROM  
X:\Program Files\PAA\Overlord3\Commands  
BEFORE IT HAS BEEN REMOVED FROM THE COMMAND SETUP SCREEN.**



## Removing a Command for the configuration

### 3.1.4 Adding a storage instrument

Most automated systems have a storage facility, racks, stacks, hotel or incubator.

Overlord3 has a special setup procedure for these.

Each storage devices can be used as a single coherent unit, i.e. one type of plates in the whole device, or in **Zones**, where particular types of plate are stored in particular racks in the storage device.

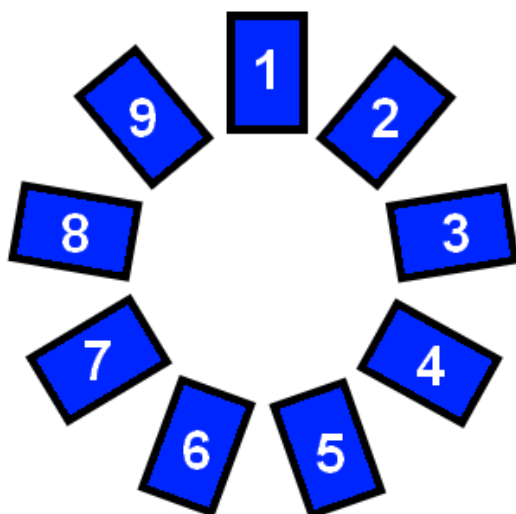
Download a Thermo Cytomat incubator 6000 instrument driver from [paa.helpserve.com](http://paa.helpserve.com) (Section 3.1.1, Getting instrument commands) and save the driver in the **Overlord3™** folder structure.

Load the instrument driver into **Overlord3™** (Section 3.1.2, Adding Instrument Commands, Selecting the storage instrument).

The user will be asked to describe the 1<sup>st</sup> zone in the plate storage device.

The first example is whether the entire incubator is one zone, i.e. all plate are the same and are treated the same by the **Overlord3™** (The whole storage device as one zone). This is represented by the entries in the table. Give the storage device a **Name**. The storage zone **Start** position is plate **1** and the **No Stacks** is **9**. Enter the **Capacity** of the stacks as 21 and Overlord3™ calculates the **End** is position as plate **189**. The **Sequence** of the plate stacks is normally 1,2,3,4,5,6,7,8,9. With default system this will apply to all plates (Add a single zone to a Thermo Cytomat 189 plate store).





**The whole storage device as one zone**

A screenshot of the 'Add PositionSet - Thermo Cytomat' dialog box. The dialog has a title bar with a close button. It contains the following fields and controls:

- Name:** A text box containing 'Plate store'.
- Start:** A spin box set to 1.
- End:** A spin box set to 189.
- No. Stacks:** A spin box set to 9.
- Capacity:** A spin box set to 21.
- Sequence:** A text box containing ',3,4,5,6,7,8,9' with a hint '(e.g. 1,2,3)'.
- Use relief positions:** An unchecked checkbox.
- Labware Types:** A list box with 'All' selected.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

**Add a single zone to a Thermo Cytomat 189 plate store**

Selecting **OK** will give the two instruments installed on the system (Selecting the storage instrument):

A screenshot of the 'Add Command' dialog box. It features a table with three columns: 'Command Name', 'Type Name', and 'Location'. Below the table is a note and two buttons.

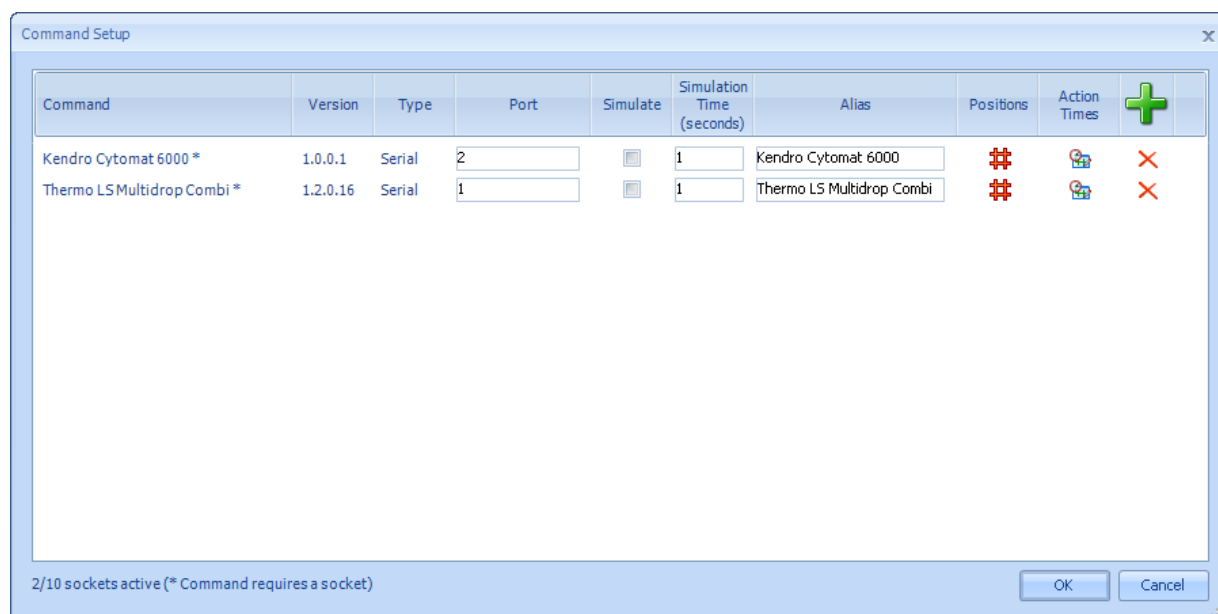
Command Name	Type Name	Location
<input checked="" type="checkbox"/> Thermo Cytomat	Overlord.Commands.ThermoCytomat.Command	D:\Program Files\PAA\Overlord3\Commands\Overlord.Co...
<input type="checkbox"/> Thermo LS Multidrop Combi	Overlord.Commands.ThermoMultidropCombi.Co...	D:\Program Files\PAA\Overlord3\Commands\Overlord.Co...

Check boxes to select multiple commands, then click OK. Double-click on a row to select a single command.

Buttons: OK, Cancel

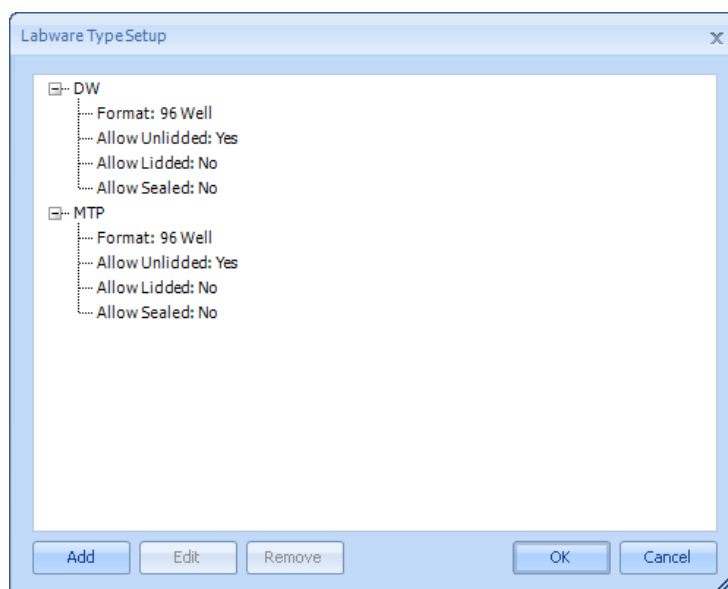
**Selecting the storage instrument**

Both instruments will be available in the **Devices** palette (Storage in Devices palette). Choose the setup option for the Thermo Cytomat instrument.



## Thermo Cytomat installed

If the procedure requires multiple plate types, e.g. 2 stacks of source plates (DW = deep well) and two stacks of assay plates (MTP = standard plates). The first thing to do is assign the new plate types in the Labware setup up tool (Section 3.2 Labware Setup). Use the **Labware Types Setup** to enter the two required plate types. On addition, the Labware type dialog box will have two options (Required labware types).



## Required labware types

Select **OK**, then assign the plate type to the storage device with the **Add** button. You will need to assign both the DW plates (Assign Source plates to storage device) and the MTP plates (Assign Assay plates to storage device) to the storage device.

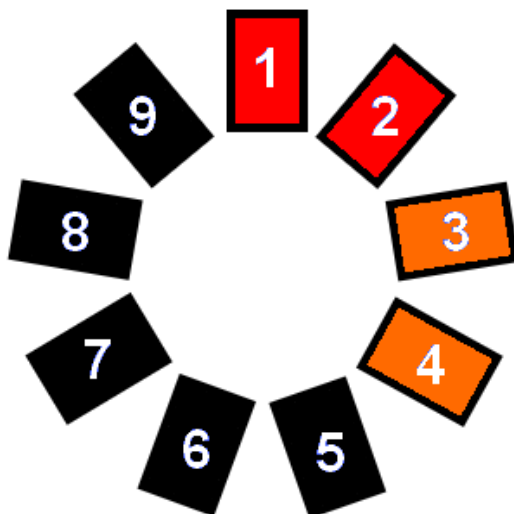
**Assign Source plates to storage device**

**Assign Assay plates to storage device**

Both types of plates are defined in the Labware Setup dialog box and are then set up and ready to use (Assigned Labware types):

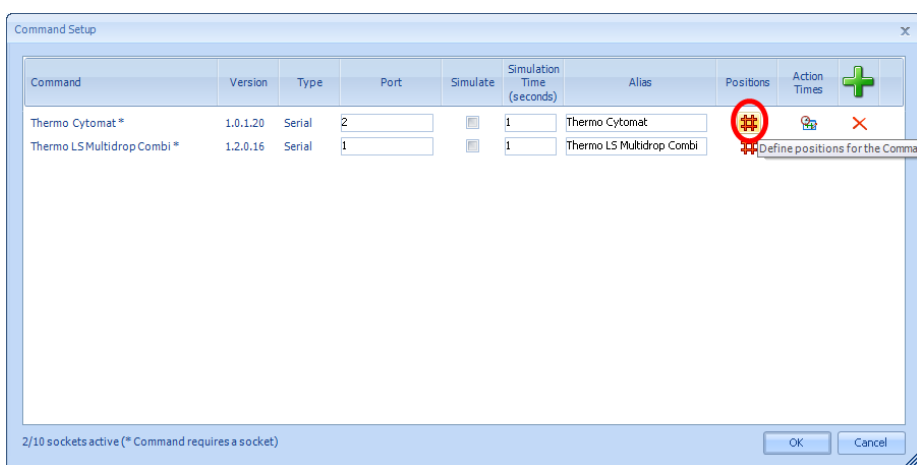
**Assigned Labware types**

The physical layout of the storage system would be as below (Two different plate zones in the storage device):



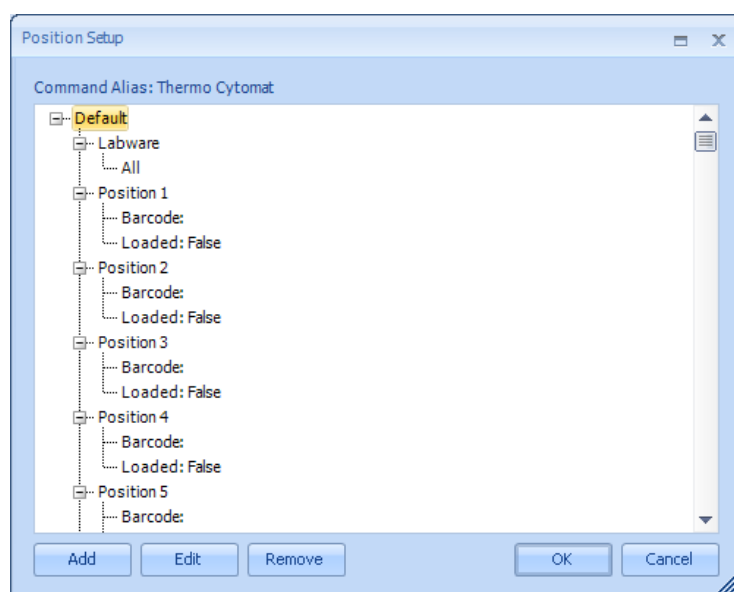
Two different plate zones in the storage device

First remove the existing layout of one zone of identical plates. **Overlord3** main window followed by **Command Setup** (top left tool bar option). Select the red hash to edit the existing positions:

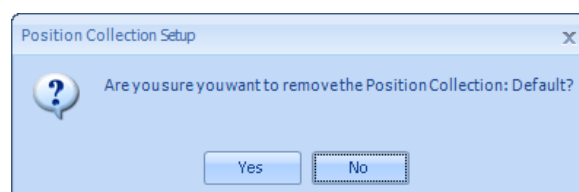


**Redefining the zones in the storage device, select the storage device**

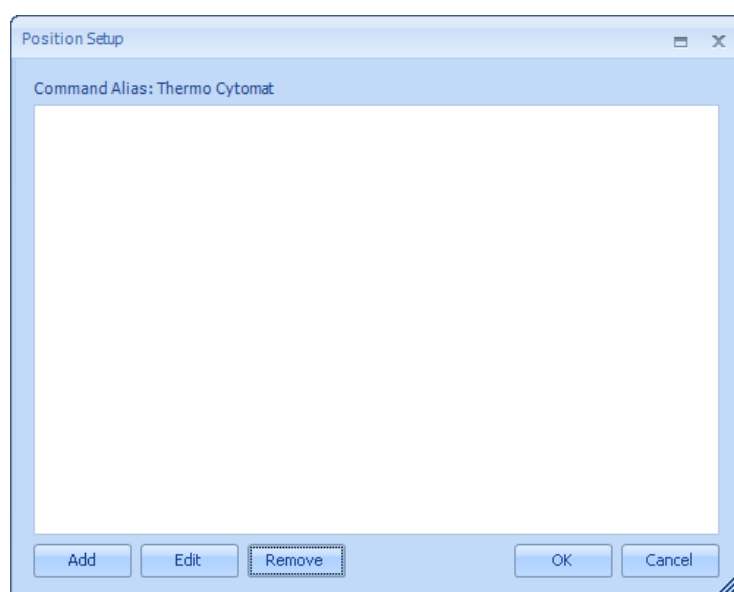
The current zoning will be defined, select **Default** and click **Remove** (Removing the existing setup), the user is asked if this is the required activity (Check if this is the required action), then if **Yes** is selected, the current zone setup is cleared (Clear zone setup for storage device):



**Removing the existing setup**



**Check if this is the required action**



**Clear zone setup for storage device**

To implement the two zone setup, select the **Add** button for the 1<sup>st</sup> zone, and enter

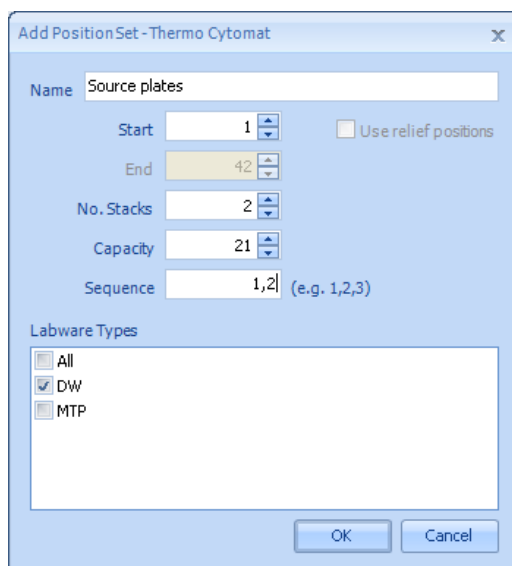
**Start = 1**

**No Stacks = 2**

**Capacity = 21**

**Sequence = 1,2**

tick the DW plate type (Set up source plate zone):



Add PositionSet - Thermo Cytomat

Name: Source plates

Start: 1

End: 42

No. Stacks: 2

Capacity: 21

Sequence: 1,2 (e.g. 1,2,3)

Use relief positions: ☐

Labware Types:

- ☐ All
- ☒ DW
- ☐ MTP

OK Cancel

**Set up source plate zone**

Select **OK** and the **Add** another zone and enter the assay plate zone information:

**Start = 1**

**No Stacks = 2**

**Capacity = 21**

**Sequence = 3,4**

Tick the MTP plate type (Zone 2 for Assay plates):

The 'Add Position Set - Thermo Cytomat' dialog box is shown. It has a 'Name' field set to 'Default'. Below it are fields for 'Start' (1), 'End' (42), 'No. Stacks' (2), 'Capacity' (21), and 'Sequence' (3,4). There is a checkbox for 'Use relief positions' which is unchecked. Below these fields is a section titled 'Labware Types' with a list box containing 'All', 'DW', and 'MTP'. The 'MTP' option is selected with a checkmark. At the bottom are 'OK' and 'Cancel' buttons.

## Zone 2 for Assay plates

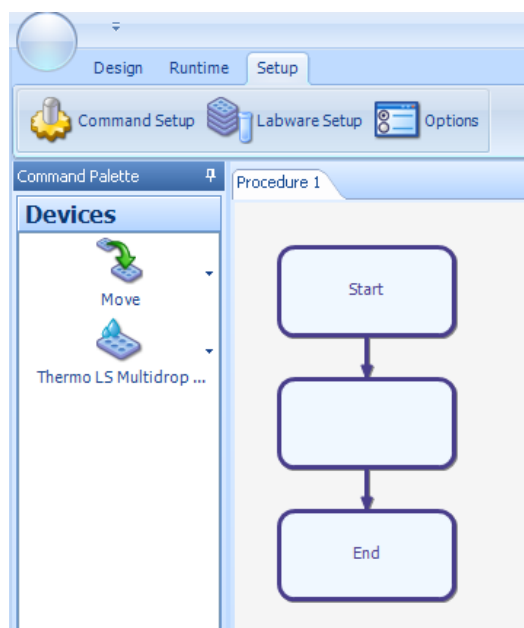
Initially the **Position Setup** dialog box only shows the details for zone 1 (View zone 1 information for Source plates). Click the – to see the 2<sup>nd</sup> zone information (View zone 2 information for Assay plates).

The 'Position Setup' dialog box is shown. The 'Command Alias' is 'Thermo Cytomat'. The tree view shows a hierarchy: 'default' (selected with a red circle) -> 'Labware' -> 'MTP' -> 'Position 1' through 'Position 5'. Each position has a 'Barcode' field and a 'Loaded: False' status. At the bottom are 'Add', 'Edit', 'Remove', 'OK', and 'Cancel' buttons.

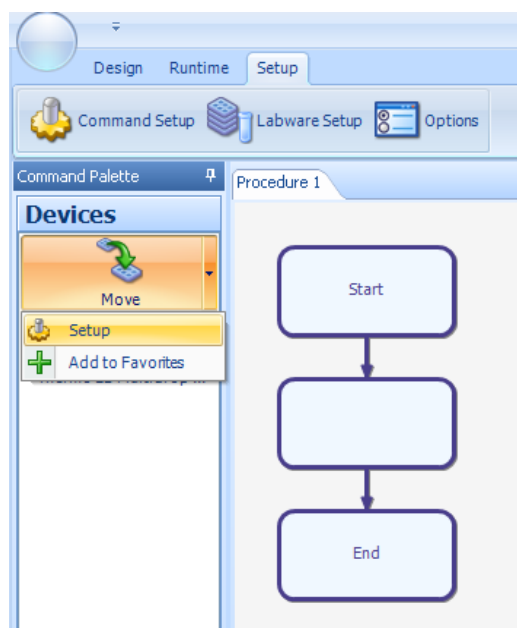
**View zone 1 information for Source plates**

The 'Position Setup' dialog box is shown. The 'Command Alias' is 'Thermo Cytomat'. The tree view shows a hierarchy: 'Default' (selected) -> 'Source plates' -> 'Labware' -> 'DW' -> 'Position 1' through 'Position 5'. Each position has a 'Barcode' field and a 'Loaded: False' status. At the bottom are 'Add', 'Edit', 'Remove', 'OK', and 'Cancel' buttons.

**View zone 2 information for Assay plates**

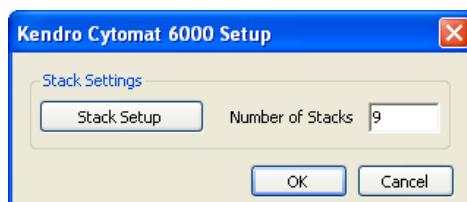


**Storage in Devices palette**



**Setup the Storage instrument**

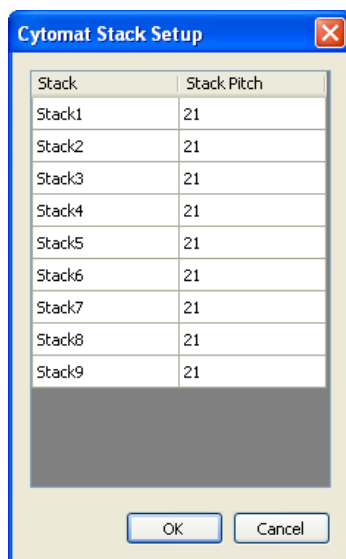
The Cytomat Setup dialog box will be displayed, enter then number of stacks installed in the hotel/incubator (Cytomat Setup dialog box):



**Cytomat Setup dialog box**

Selecting the Stack Setup button allow the setup of the individual stack pitches (Cytomat Stack Pitch dialog box). This allows deep well plates and standard plate stacks to co-exist in one storage module:

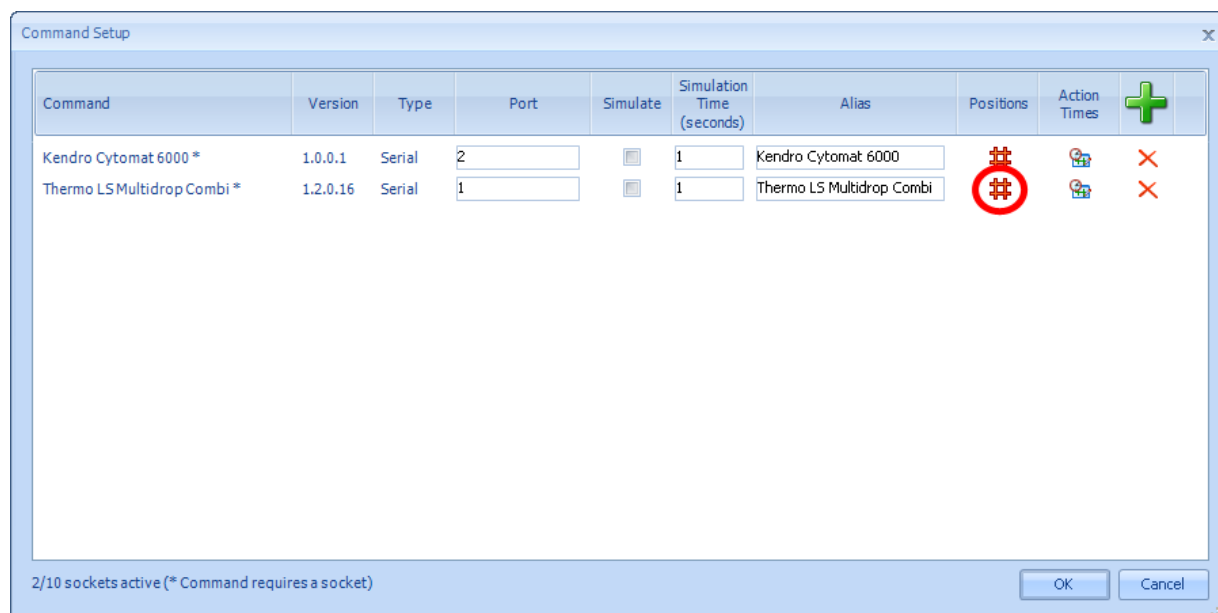


**Cytomat Stack Pitch dialog box**

### 3.1.5 Positions

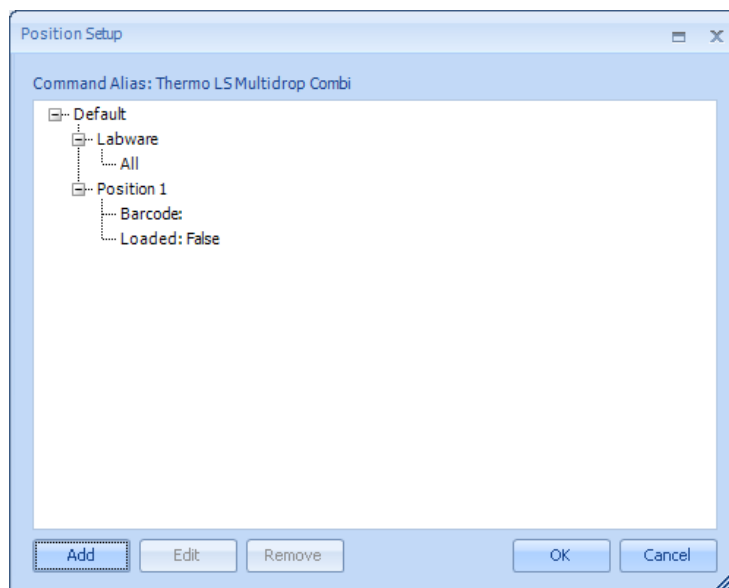
Most instruments only have one position for a robot to place a microplate. Other instruments do have multiple positions.

Viewing the Thermo Multidrop position configuration (Configuring the number of positions on an instrument):



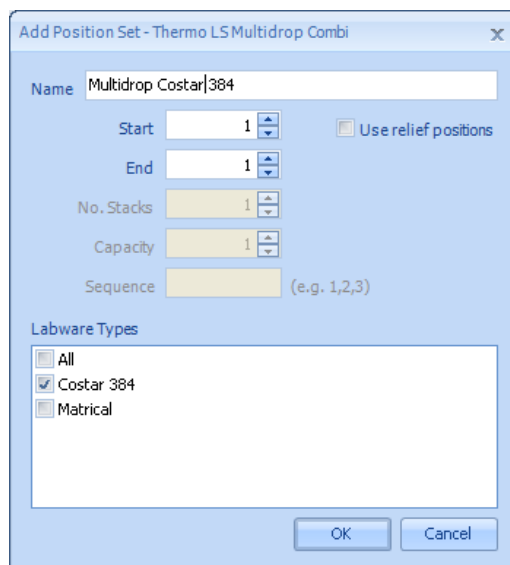
#### **Configuring the number of positions on an instrument**

The default configuration is shown (Default position information for a Thermo Multidrop). The single plate position can be seen.:



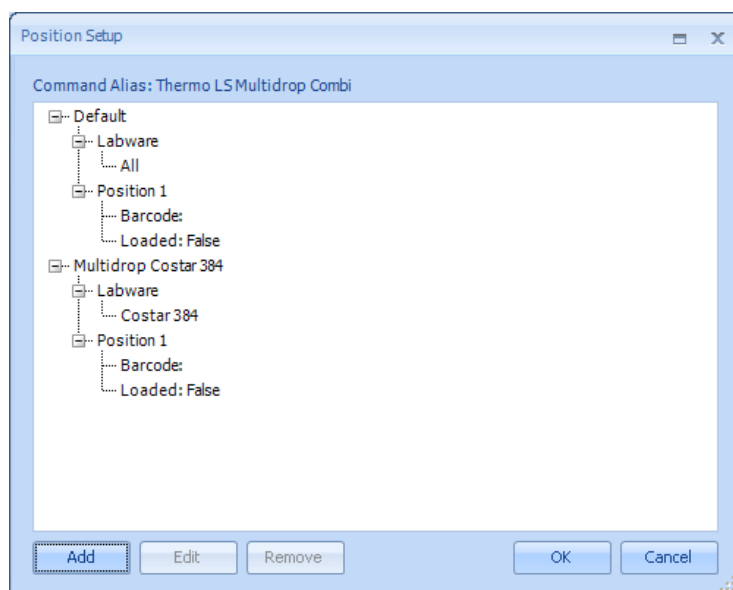
**Default position information for a Thermo Multidrop**

By selecting the **Add** button, the user can enter particular uses of the instrument (Add a new position on a Thermo Multidrop instrument):



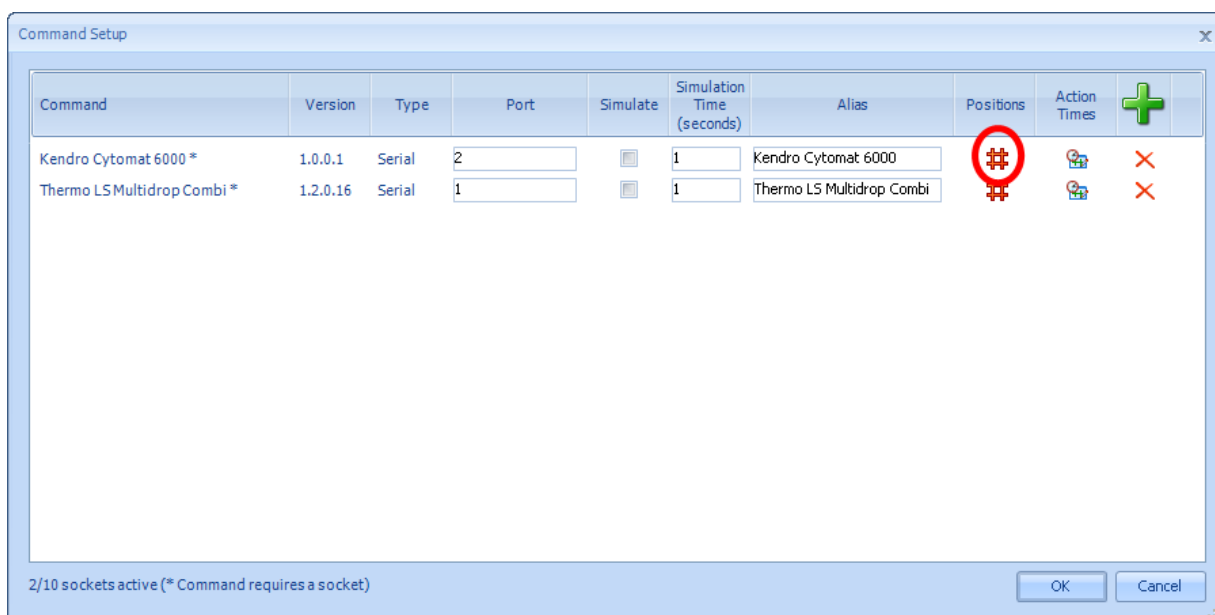
**Add a new position on a Thermo Multidrop instrument**

Select OK to exit, to view all positions on Thermo Multidrop instrument (All positions on Thermo Multidrop instrument):

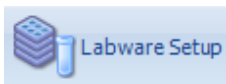


**All positions on Thermo Multidrop instrument**

For a storage device, select that positions option (Position information for storage device).

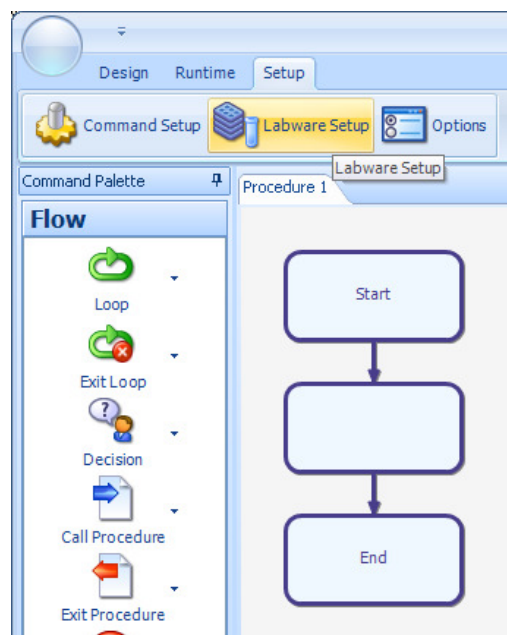


**Position information for storage device**



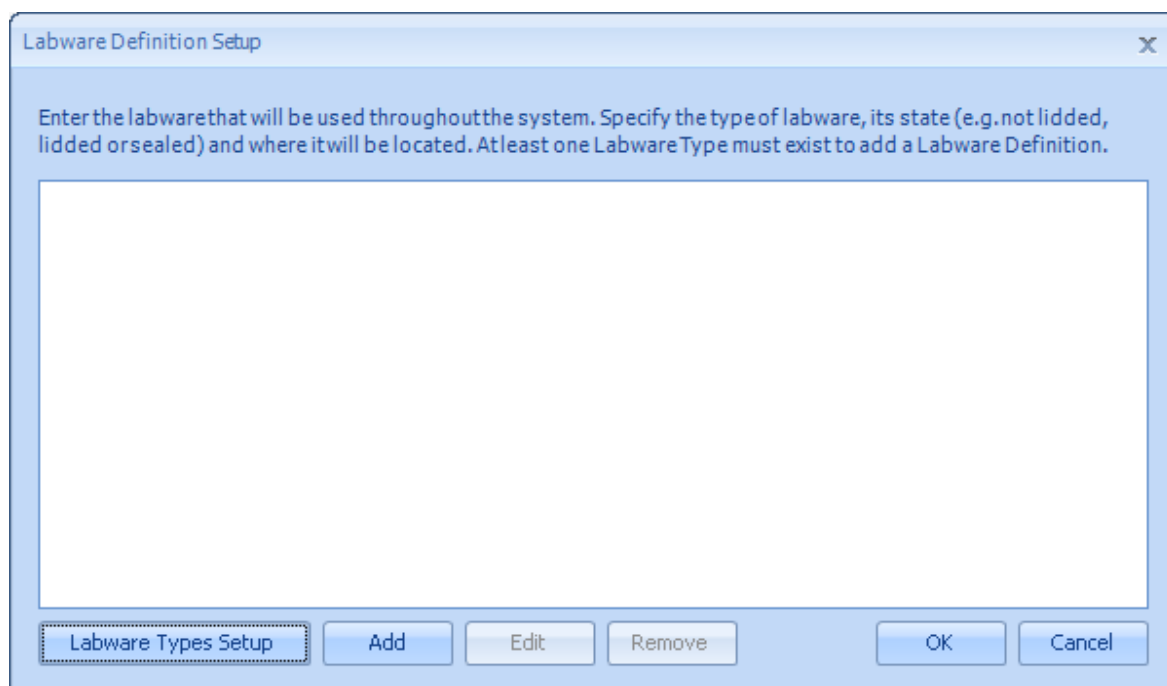
## 3.2 Labware Setup

The location, type and size of labware are a key components to the automated setup options of Overlord3. The Labware Setup command can be accessed from the Setup, Labware Setup tab option (Labware Setup tab option). This section also allows the user to allocate a plate type to a storage location:



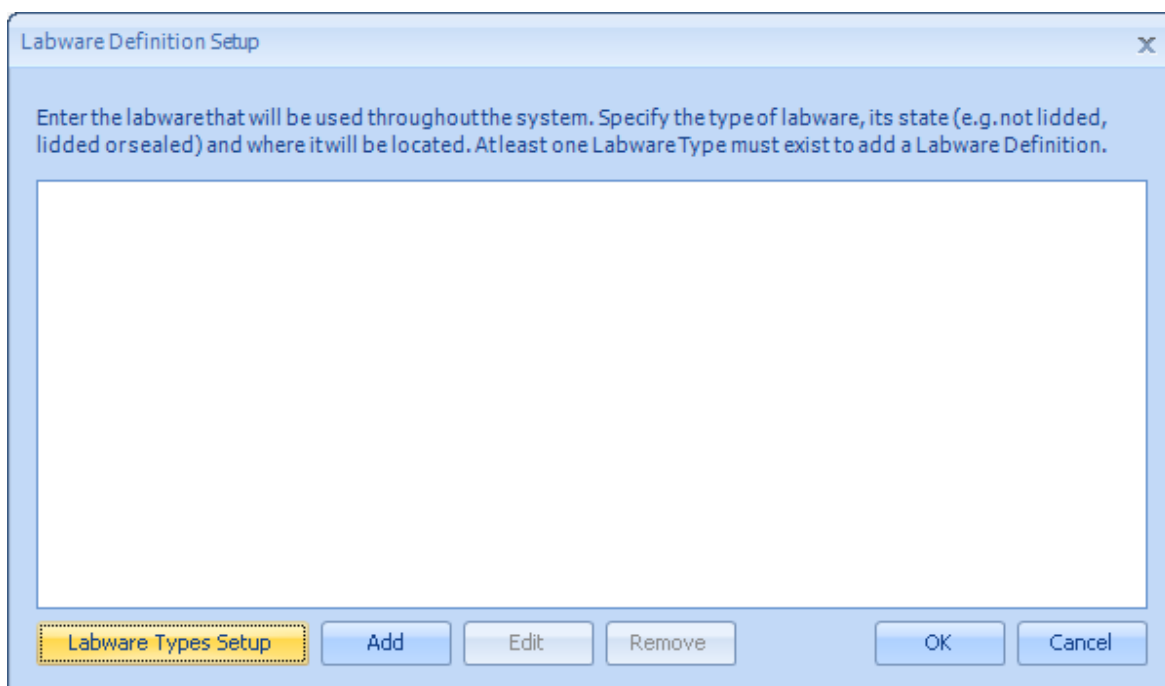
**Labware Setup tab option**

This will present the Labware Definition Setup dialog box (Labware Definition Setup dialog box).



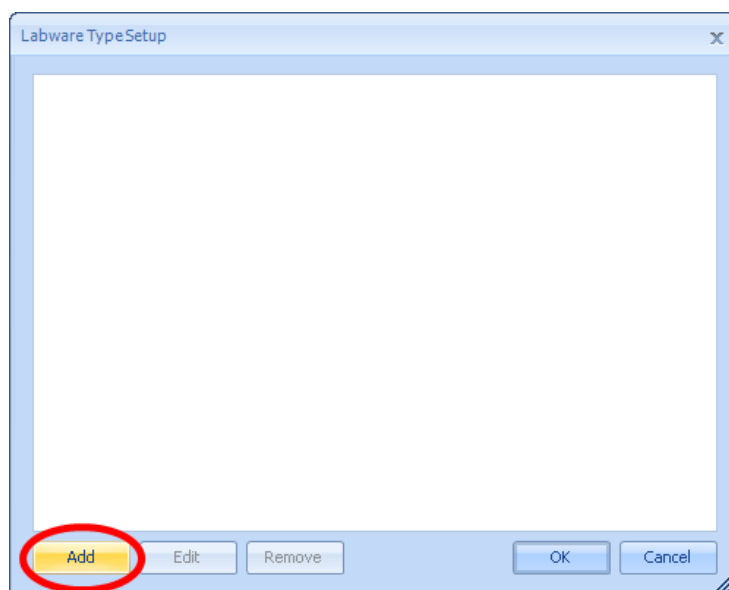
**Labware Definition Setup dialog box**

To setup a new labware setup type, choose the **Labware Types Setup** button (Labware Type Setup button).



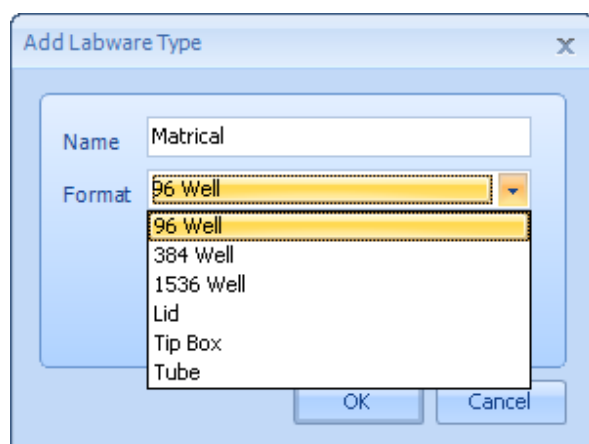
**Labware Type Setup button**

Then to add a new type, choose **Add** from the **Labware Type Setup** dialog box (Add new labware type).

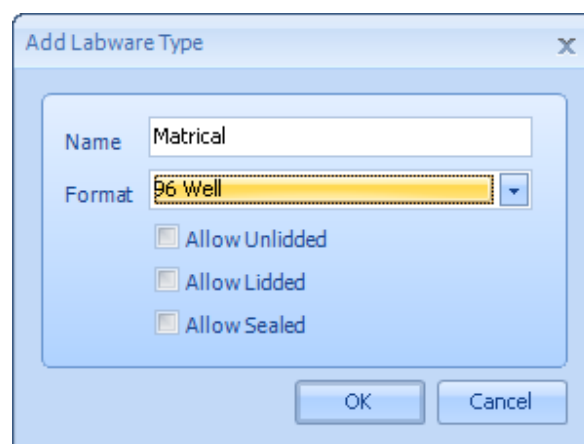


## Add new labware type

The new labware can be named, the type of labware selected (Add labware type name and number of wells) and any features such as lid or sealed can be selected (Choose plate type options).

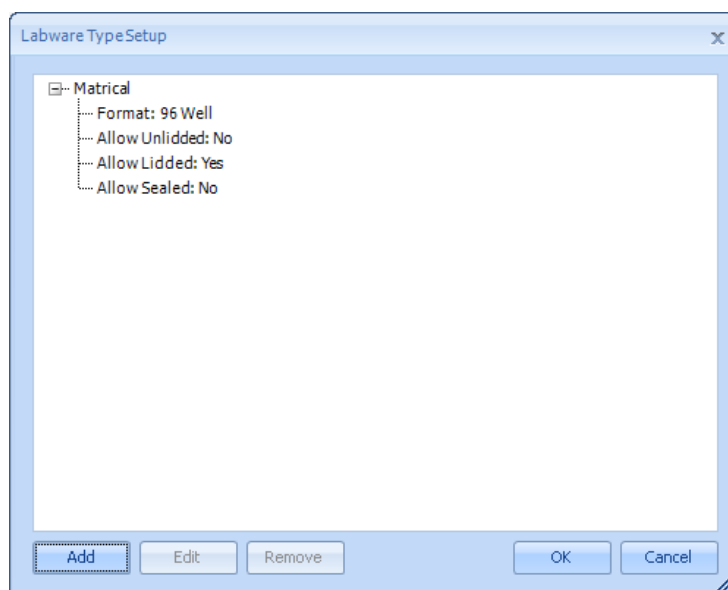


## Add labware type name and number of wells



## Choose plate type options

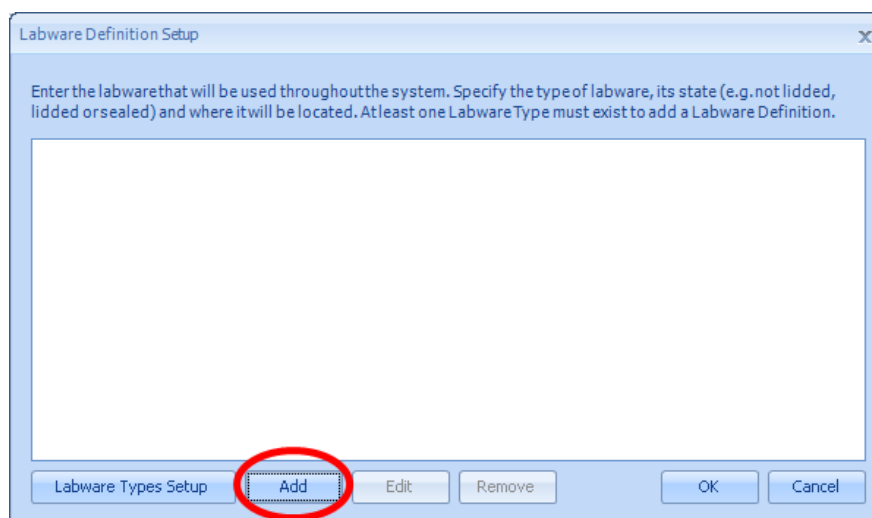
Selecting OK, will display the features of the new plate type (New plate type added to library):



#### New plate type added to library

The plate type can be changed with the **Edit** button or removed with the **Remove** button. Once all the new plate types have been added, the user can return to the Labware Definition Setup dialog box by selecting **OK** (Labware Type Setup button).

The labware can now be allocated to a storage location, select the **Add** button (Add the plate type to a storage location):

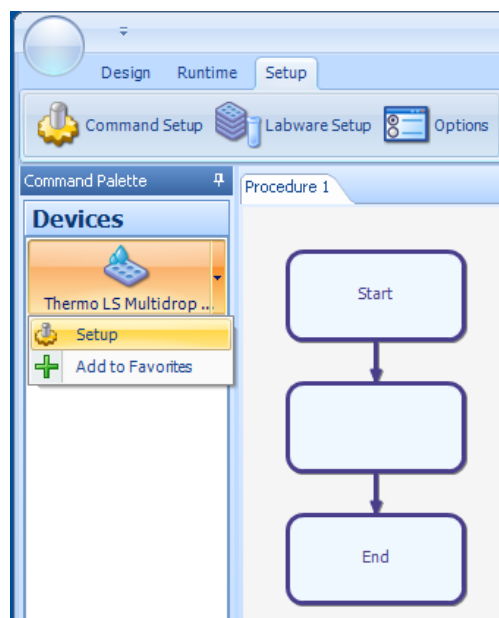


#### Add the plate type to a storage location

### 3.3 Command Setup

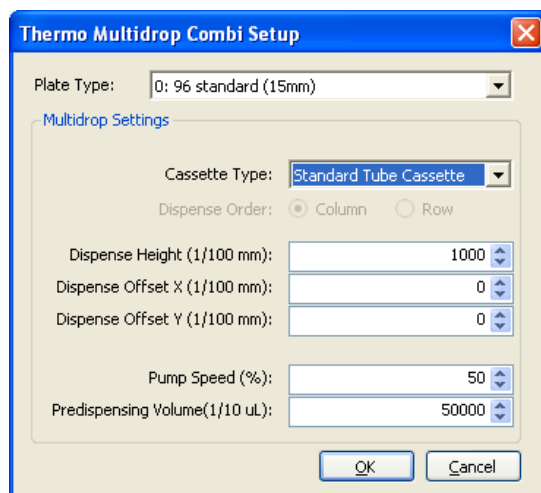
Some commands have setup dialogs available for setting parameter specific to that command. Typical parameters are connection timeout values, the location of data files and the location of third party software executables. Setup also brings up the Teach software for robotic arms and diagnostic software.

To access the Setup section of a command, click on the small arrow to the right of the command in the **Devices** palette. This will bring up a small context menu and if the **Setup** is available, it will be listed as a menu entry (Instrument command installed in Devices palette):



**Instrument command installed in Devices palette**

Click on this to bring up the **Setup** dialog box (Thermo Multidrop setup screen):

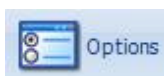


**Thermo Multidrop setup screen**

Default instrument settings can be modified in this dialog box.

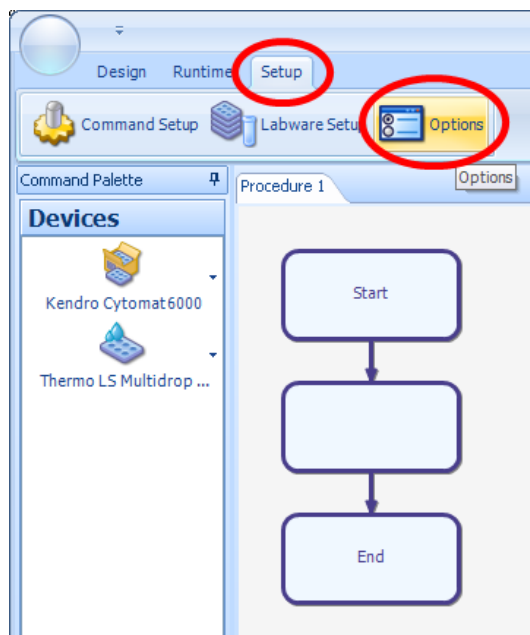
Refer to the Readme.txt file that can be found for each command on the [paa.helpserve.com](http://paa.helpserve.com) site for more information on what settings can be altered. Information can also be found in the command help file if it has been installed.





## 3.4 Options

There are various system settings for Overlord3 that are accessible by clicking on the toolbar and selecting **Setup** and clicking the **Options** button



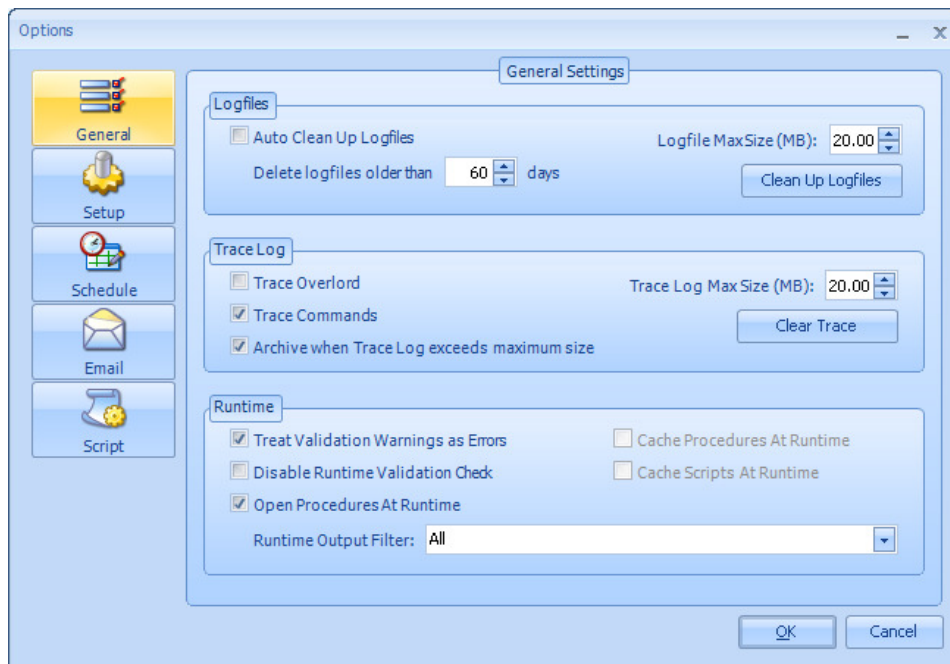
**Options Button**

There are five sections in Options:

- ▶ General
- ▶ Setup
- ▶ Schedule
- ▶ Email
- ▶ Script

### 3.4.1 Options, General

This section allows the user to control the **Logfiles**, **Trace Log** and **Runtime** parameters (Options, General).



**Options, General**

### 3.4.1.1 Logfiles

Each and every time that **Overlord3™** is run, a log file is produced. Logfiles are stored in folder **X:|Program Files|PAA|Overlord3|Logfiles** (where **X** is the **Overlord3™** installation directory).

Over time, these log files can occupy a very large amount of PC hard disc space. The user can move, archive or delete these files manually, or **Overlord3™** can be setup to automatically delete the logfiles based on the date that the log file was written. Tick the **Auto Clean Up Logfiles** check box to automatically delete logfiles older than the value in **Delete logfiles older than ... days** every time **Overlord3™** is started. The user can delete unwanted logfiles NOW, by selecting the file age in **Delete logfiles older than ... days** data entry box and clicking **Clean Up Logfiles**.

**Logfile Max Size (MB)** specifies the maximum size of an individual logfile. The first logfile for a run will be named by the date and time with **\_\_1** appended on the end. If the run generates a logfile that is over the maximum value a new logfile is created with same data and time as the first file, but with **\_\_2** on the end.

### 3.4.1.2 Trace Log

The **Trace Log** is usually activated to identify system problems. Selecting **Trace Log** options write detailed information into a file to assist with problem solving and error detection.

Selecting **Trace Overlord** writes low level information on internal Overlord3 processes. The information provided by this trace function is only required in exceptional circumstances and should only be turned on if requested by a member of the PAA Technical Support Team.

Selecting **Trace Commands** writes the low level communications between Overlord3 and the instrument, this is usually used to determine if Overlord3 is sending the wrong commands to the instrument, or the instrument is replying in an unexpected way.

The usual configuration for a system is to have **Trace Overlord Off** and **Trace Commands On**

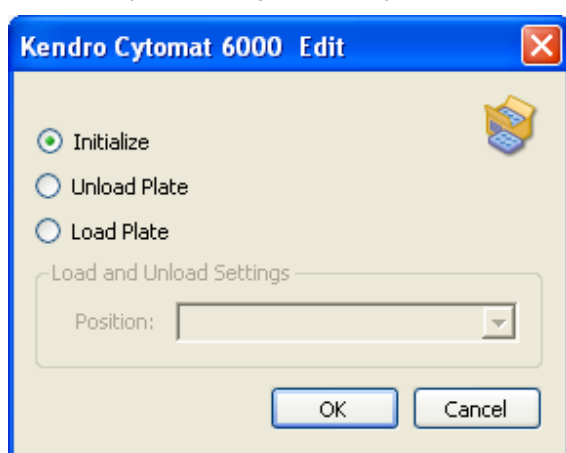
The **Trace Log** can get very large, so a maximum size can be set in the **Trace Log Max Size (MB)** data entry box. This log can be archived rather than deleted with the **Archive when Trace Log exceeds maximum size** check box. Or the trace log can be cleared NOW with the **Clear Trace** button.

## 3.4.1.3 Runtime

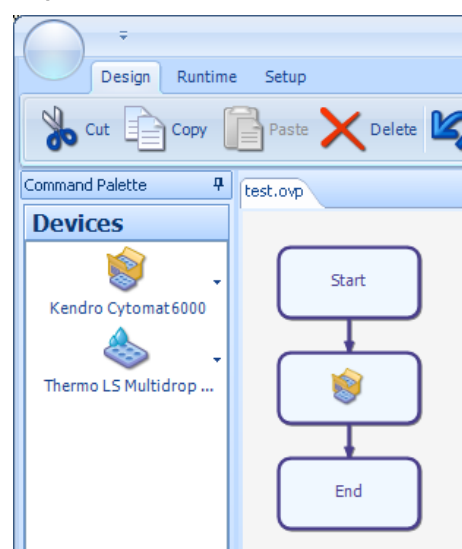
### Treat Validation Warnings as Errors

At the beginning of a run, Overlord3 does a quick check to see if the procedures to be actioned have any structural errors. If the **Treat Validation Warnings as Errors** check box is selected, the validation stage will stop displaying the errors, unselected, the warnings are just highlighted.

For example, if an initialise Cytomat command (Initialise Cytomat instrument dialog box) is put in an **Overlord3** procedure (Initialise Cytomat instrument procedure).

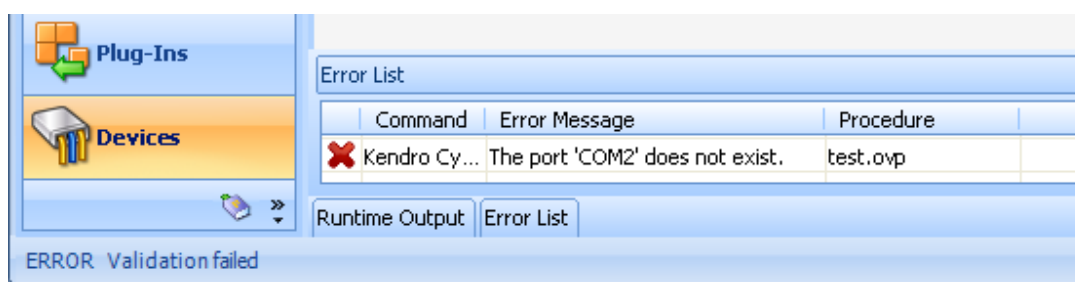


Initialise Cytomat instrument dialog box



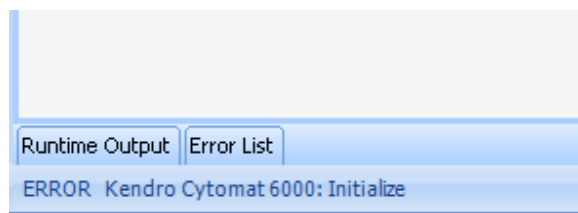
Initialise Cytomat instrument procedure

If the **Treat Validation Warnings as Errors** check box is selected, when the procedure is run, the error will be detected in the validation phase of the run (Error on validation failure) and the procedure will not be actioned:

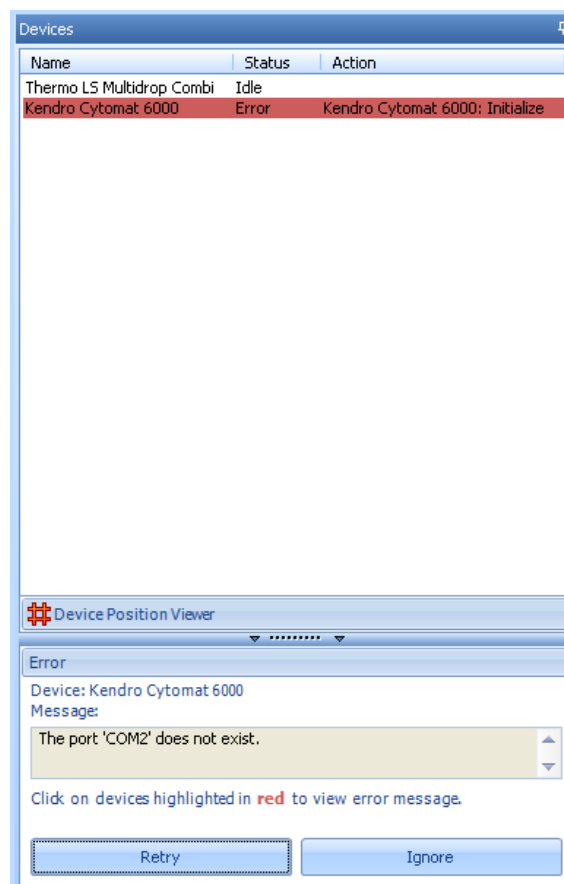


Error on validation failure

If the **Treat Validation Warnings as Errors** check box is unselected, the error will be detected at runtime ( Runtime error detected (Overlord3 screen, bottom left) and Runtime error detected, (Overlord3 screen bottom right)). It is necessary to hit the **Retry** or **Ignore** to continue:



**Runtime error detected (Overlord3 screen, bottom left)**



**Runtime error detected, (Overlord3 screen bottom right)**

## Disable Runtime Validation Check

Checking the **Disable Runtime Validation Check** box overrides the setting of **Treat Validation Warnings as Errors**, so errors are only reported at runtime.

## Open Procedures At Runtime

Checking the **Open Procedures At Runtime** box, allows all the sub-procedures to opened and seen as the sub procedures run.

## Cache Procedures At Runtime

If the option is selected procedure files are saved in memory at runtime. The standard behaviour is to open the procedure from disk every time. This option should be used on finished or final sytems to improve performance.

## Cache Scripts At Runtime

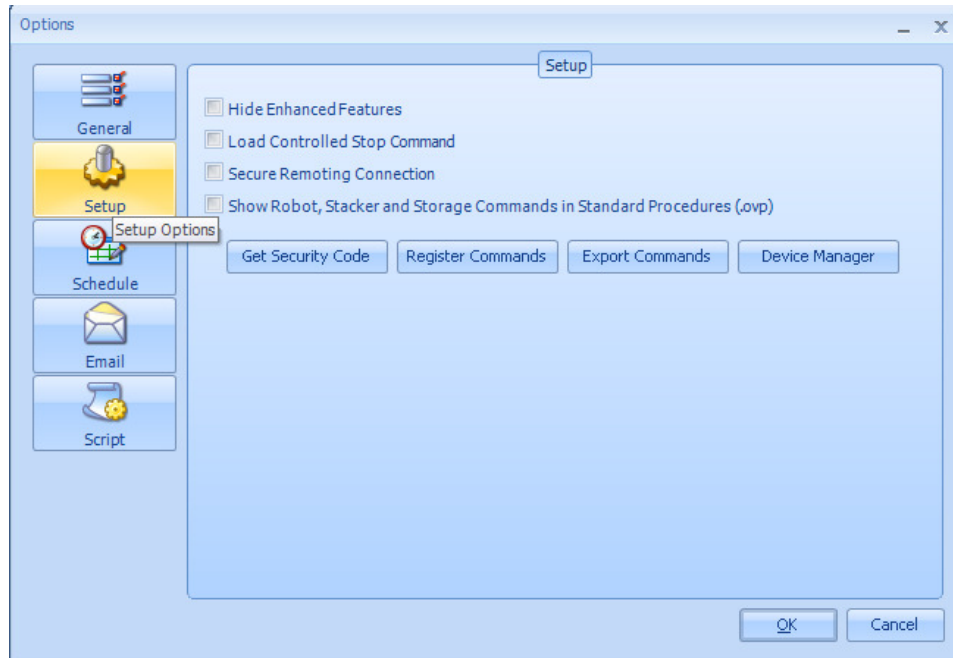
If the option is selected script files are saved in memory at runtime. The standard behaviour is to open the script from disk every time. This option should be used on finished or final sytems to improve performance.

## Runtime Output Filter

Limits the type of entries displayed in runtime output screen at runtime.

## 3.4.2 Options - Setup

These options effect the setup and **Overlord3™** display features:



**Options, Setup**

### Hide Enhanced Features

Enhanced features refer to the upgrade of an additional runtime engine to upgrade **Overlord2** to **Overlord3**. With this selected, **Overlord3** performs at edit and runtime the same as **Overlord2**, i.e. only real-time, event-driven schedules are available. Unselected, the user is able to edit and run dynamic time-resolved schedules.

### Load Controlled Stop Command

The Controlled Stop Command is a special Command used by Assay Manager and third-party integrations that use the **Overlord Framework**. It is not required by stand-alone **Overlord3** installations.

### Secure Remoting Connection

This option is only required when using **Overlord Remoting**. In most cases **Overlord Remoting** will run without this option checked. However, in rare circumstances, the security settings on the network require this option can be selected. In this case the option must be turned on in both **Overlord3** and the **Remote Listener**.

### Show Robot, Stacker and Storage Commands in Standard Procedures (.ovp)

By default any Robot, Stacker or Storage devices are hidden in the Command Palette when building Standard Procedures (.ovp). This is to guide users to use the Move Command. Select this option to show hidden device types in the Command Palette when creating Standard Procedures.

### Get Security Code

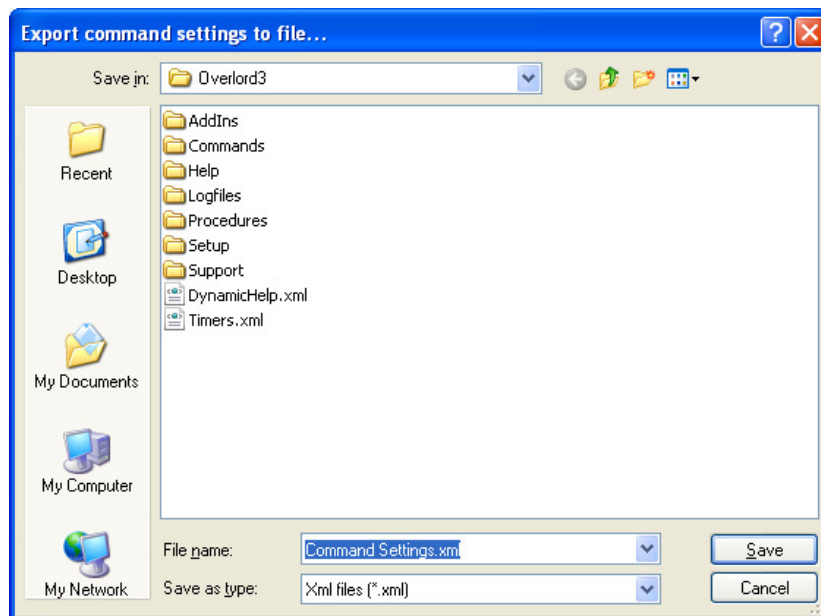
The activation of **Overlord3** is described in a later section (Section 3.5 Overlord3™ Activation).

## Register Commands

The registration of Overlord1 compatible commands for use with **Overlord3™** is described in a later section (Section 3.6 Overlord1 Instrument Driver registration).

## Export Commands

The Export Commands button will create a copy of the Command settings and save them to an XML file location that can be specified in the 'Save As' Dialog window that appears (Export Command dialog box):



**Export Command dialog box**

If there is a problem with the system, it is worth sending this generated file to PAA for technical support.

## Device Manager

Shortcut to the Windows XP Device Manager

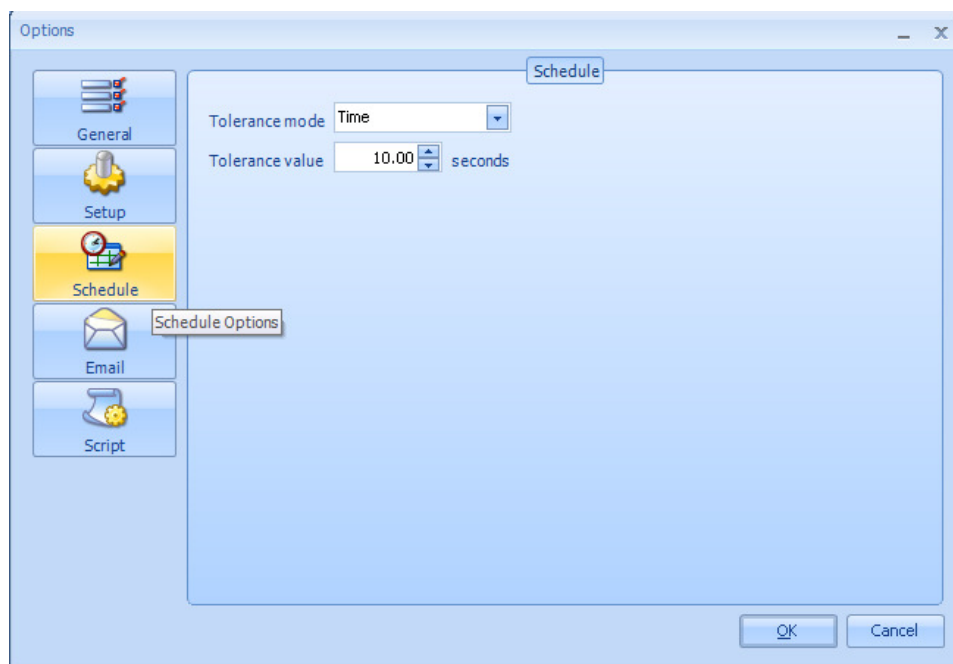
### 3.4.3 Options, Schedule

#### Tolerance mode and value (seconds or %)

The Tolerance options are used when running a **Schedule Procedure** in which a **Move Command** has the **Use wait time** option selected. The system will compare the actual wait duration with the value set in the Procedure and log any values outside the given tolerance.

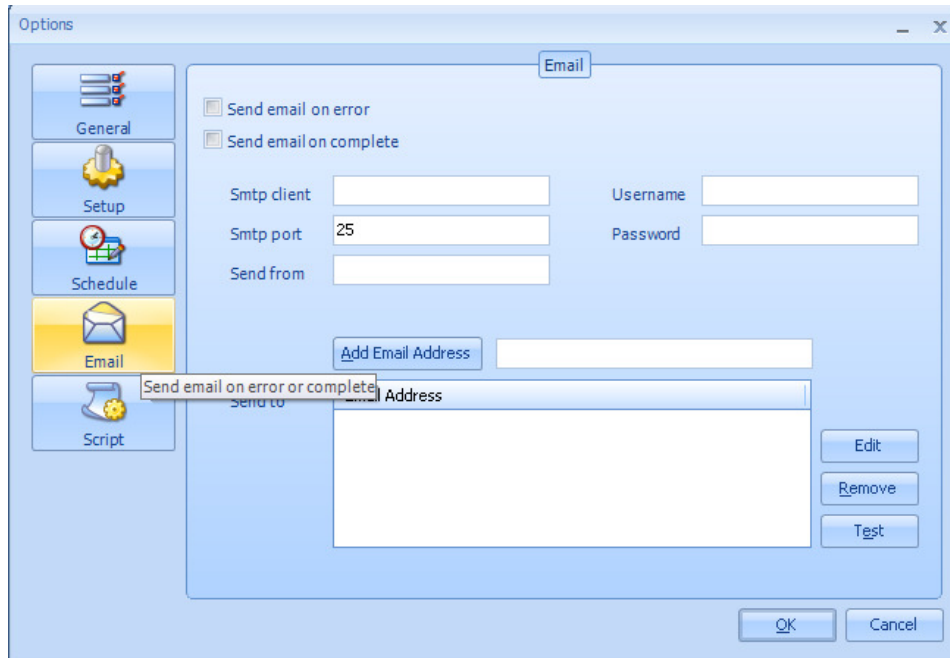
Options are **Time** and **Percentage**.

Determines whether the **Tolerance value** is used as an absolute **Time** value (e.g. plus 10 seconds) or as a **Percentage** of the set wait time (e.g. plus 5% on 600 seconds = 30 seconds):



Options, Schedule

### 3.4.4 Options, Email



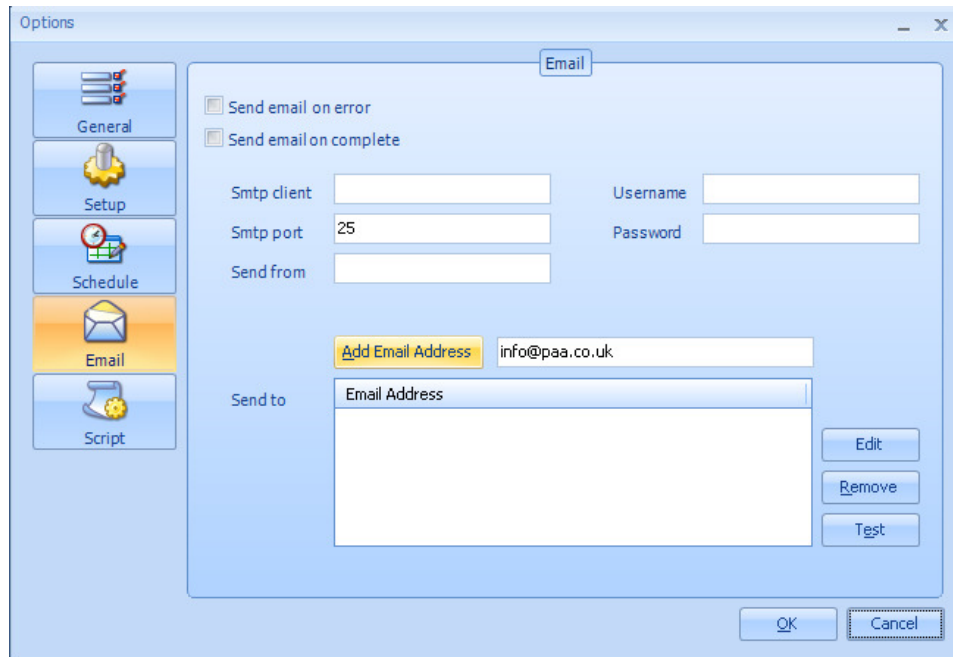
Options, Email

### 3.4.5 Email on Error

If the **Send email on error** check box is selected, an email will be sent to the listed email addresses every time there is an error with a command. The following settings need to be made for this to work.

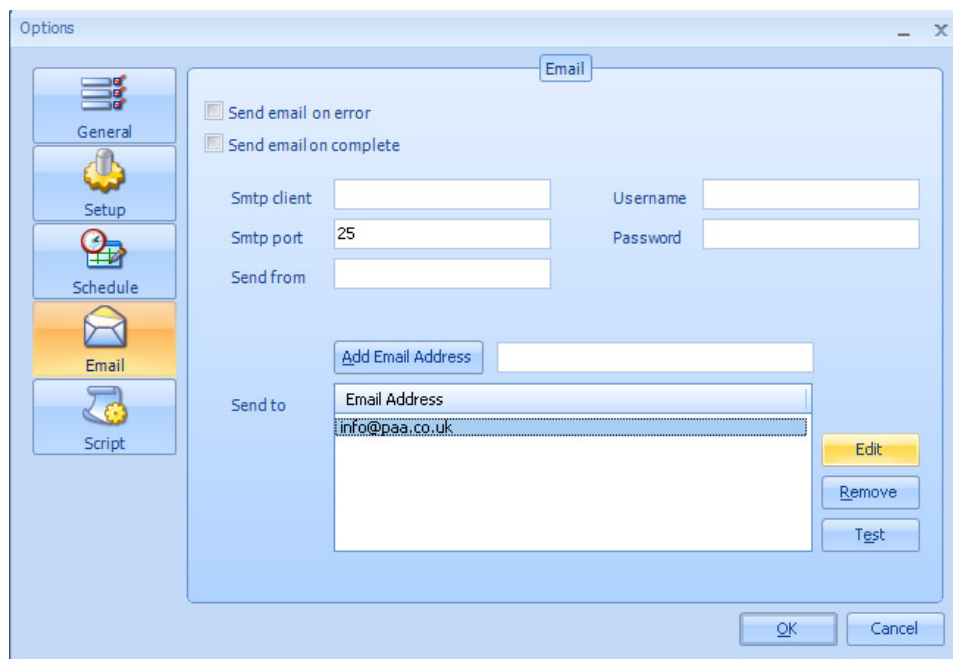
- **SMTP Client:** The SMTP Outgoing mail server address (e.g. smtp-1.mail.vi.net)
- **SMTP Port:** This number is normally 25
- **Send from:** The name that will appear on the email when it is received (e.g. Overlord3PC)
- **Username:** The username for the POP/SMTP account that is being used
- **Password:** The password for the POP/SMTP account that is being used
- **Send to:** A list of the email addresses to send the message to. These must be full email addresses (e.g. yourname@companyname.com) and are added with the **Add Email Address** button (Add Email Address):





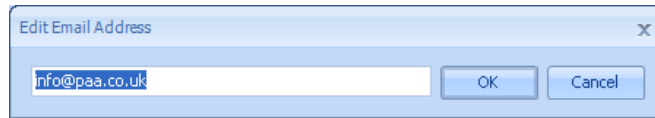
## Add Email Address

This will add the email address to the list of recipients (Email address added, edit option):



## Email address added, edit option

To edit the email address, select the email address to edit and click on **Edit** (Email address added, edit option) to bring up the edit dialog box (Edit email address):



### Edit email address

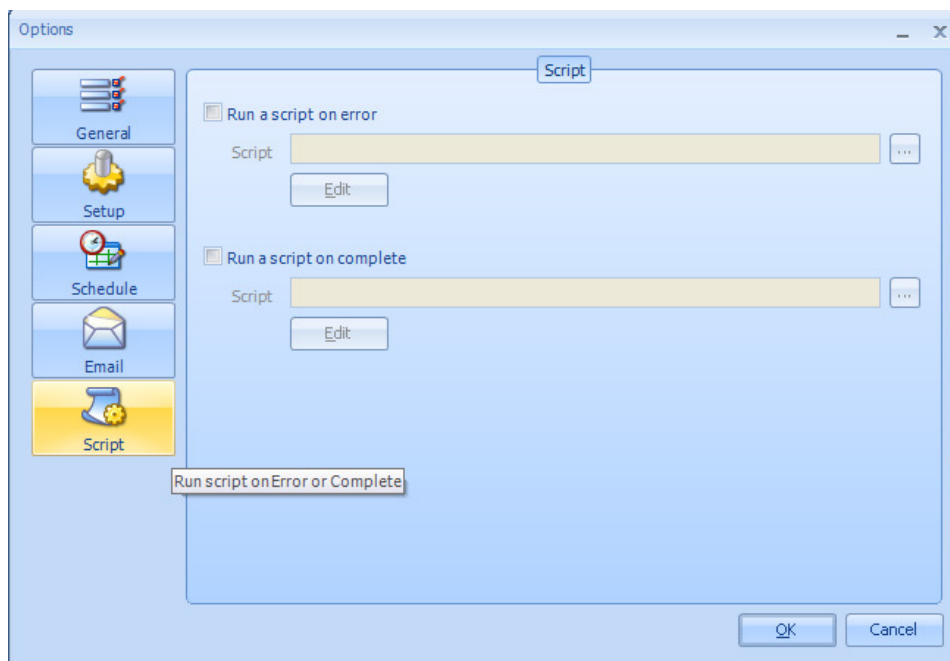
To remove an email address, select the email address and click on **Remove**.

The settings can be tested by clicking on the **Test** button. This will send an email message to the listed recipients exactly as the system would if there was an error. Any problems with the settings will become apparent at this point.

**Send an email on complete** uses the same parameters as above, except an email message is sent to the email list at the completion of an Overlord3 procedure.

## 3.4.6 Options, Script

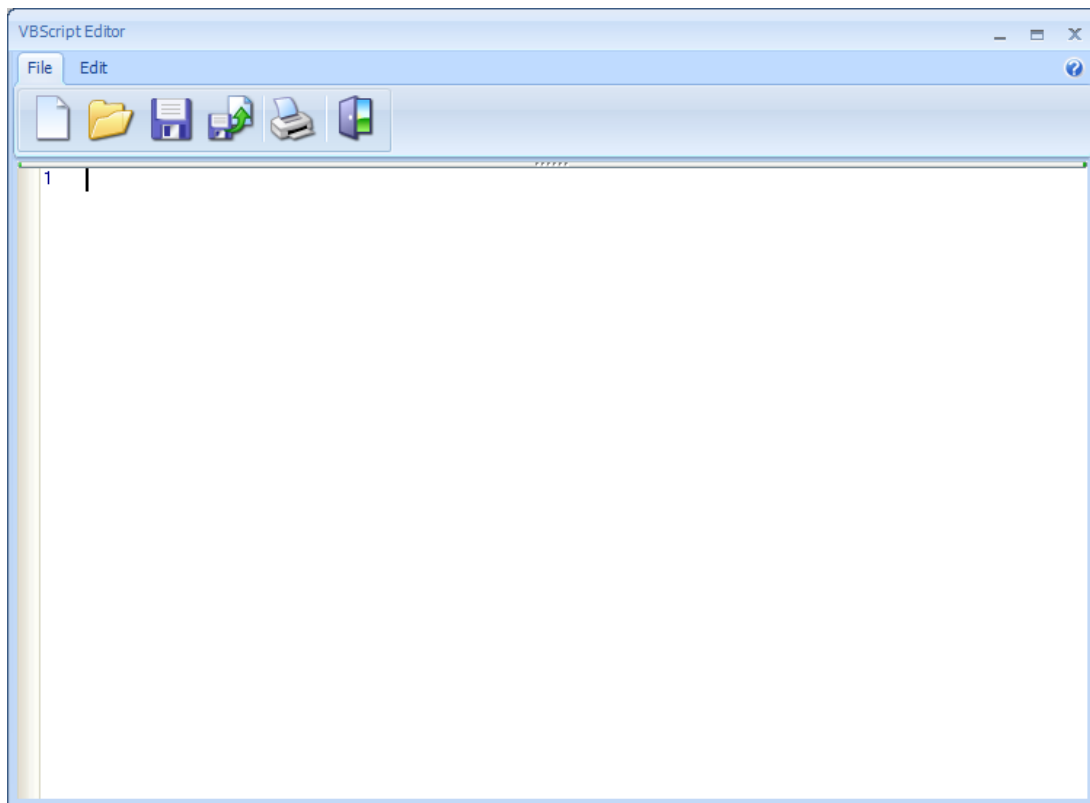
VB Scripts allows the extension of **Overlord3™** to allow novel and bespoke software modules to be used:



### Options, Script

- If the **Script on error** check box is selected, the selected script will be run when there is an error with a Command. Typically, the selected script will contain code to report the error to a data store or send a message via a messaging application (e.g. SMS). To create a script, click on the **Edit** button. This will bring up the Overlord3 Script window (VB Script editor window):

When an error is generated, these variables will be set with these error values. There is additional information relating to the implementation of VB/C#/VB.NET Scripting in the Editing section.



**VB Script editor window**

All **Overlord3™** variables are available in the scripting environment to report the error. They are

- [ErrorNumber] – The Error code of the generated error
- [ErrorDevice] – The Alias of the device that has generated the error

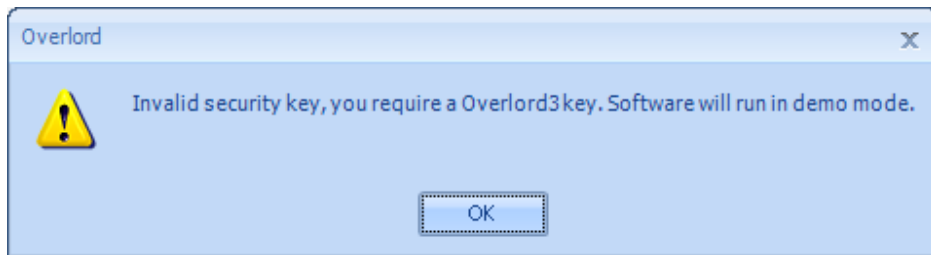
[ErrorDescription] – The error description  
**Run script on complete** runs the script at the completion of an Overlord3™ procedure.

## ***3.5 Overlord3™ Activation***

When you run a procedure (MAJOR LINK), you may see an error dialog box (No security key found) or wrong key Incorrect security key).



**No security key found**

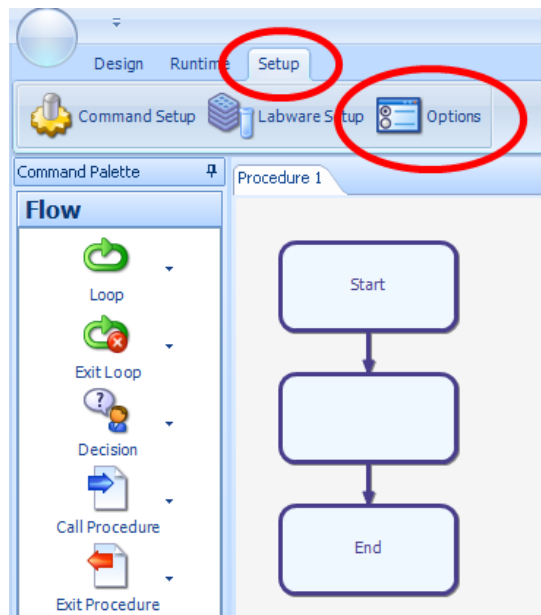


**Incorrect security key**

You need to apply to **paa** for an activation key for running the Overlord3 software.

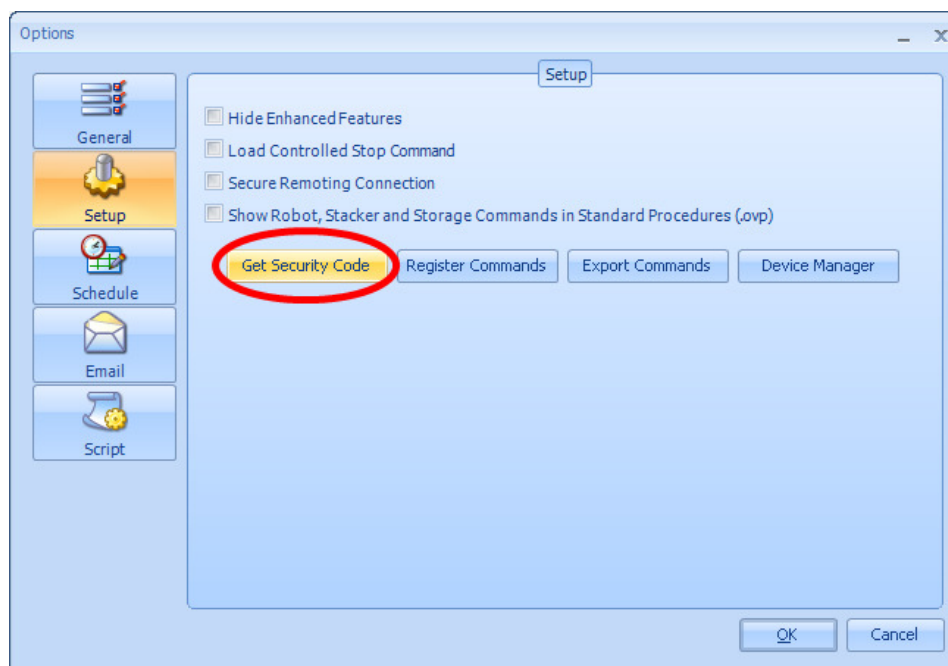
Follow the following instructions for product activation.

Run Overlord3 and select the **Setup** tab, and then the **Options** tab (Setup Options tab).



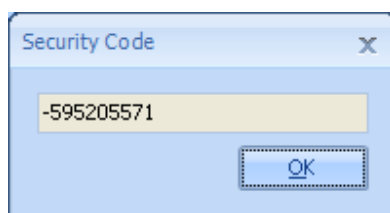
**Setup Options tab**

This will bring up the **Setup** dialog box. Select the **Setup** side tab and the **Get Security Code** button (Get Security Code button):



#### Get Security Code button

This will bring up the security code dialog box (Security Code dialog box):



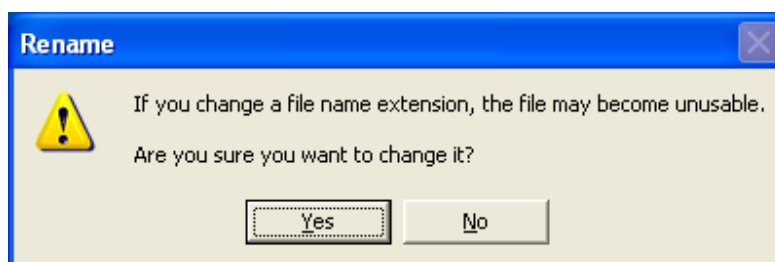
#### Security Code dialog box

Send the number by email to [info@paa.co.uk](mailto:info@paa.co.uk) and you will receive the activation code by return email. The -ve sign **IS** significant if present.

The activation code is usually sent as a text file (e.g. **activation.txt**), save it on your desktop (Text file icon), rename the file as a reg file (**activation.reg**). You will receive a warning – select yes:



#### Text file icon



#### File extension warning

You will then see the file icon change (Registry file icon). Then double click on the file name (**activation.reg**), to run the file. You may receive a security warning (Security Warning), which you should answer **Run**.

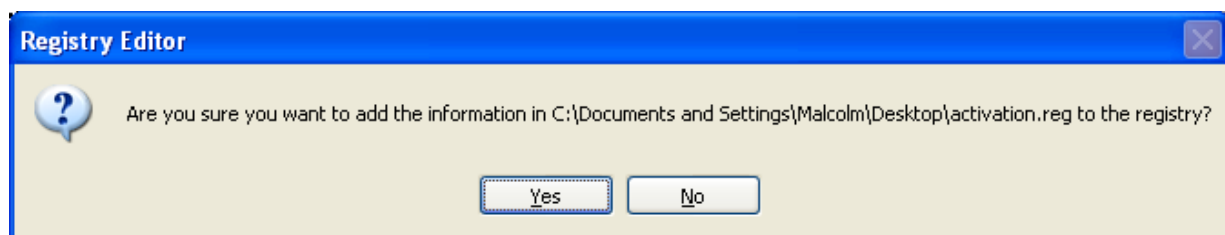
This is a single user activation key for the computer that has been activated only.



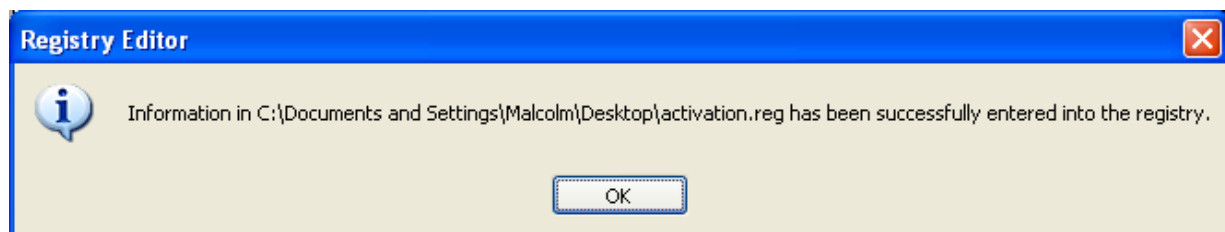
activation.reg

**Registry file icon****Security Warning**

You will see an offer to add information to registry (Add information dialog box) that you should answer **Yes**.

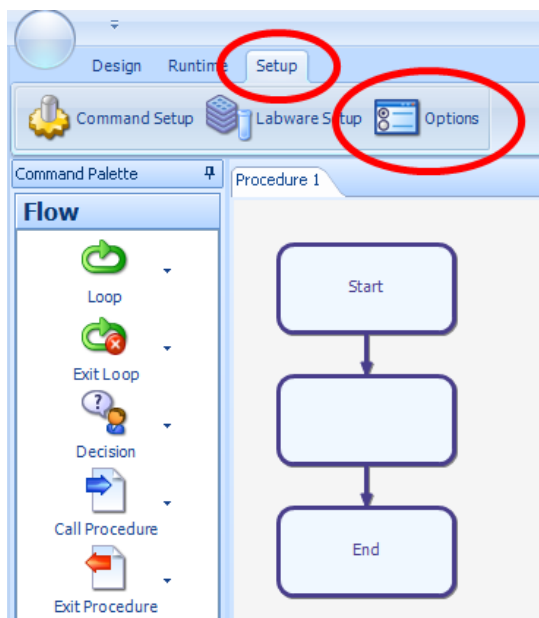
**Add information dialog box**

This will be followed by an information dialog box (Information added dialog box) that the information has been added.

**Information added dialog box**

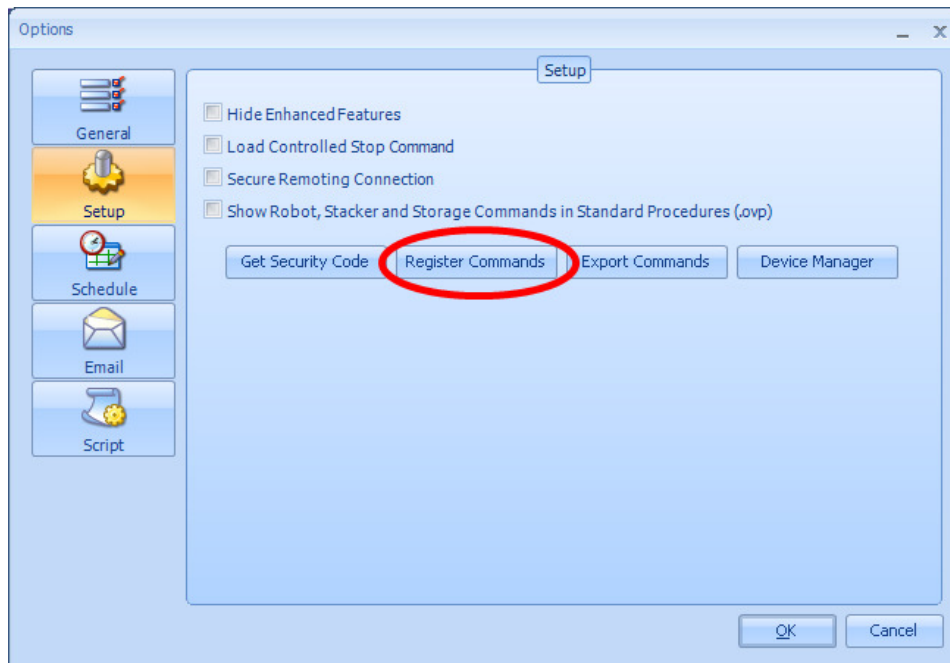
### ***3.6 Overlord1 Instrument Driver registration***

To register any new Overlord1 instrument drivers on your **Overlord3** installation, run **Overlord3** and select the **Setup** tab, and then the **Options** tab (Setup Options tab).



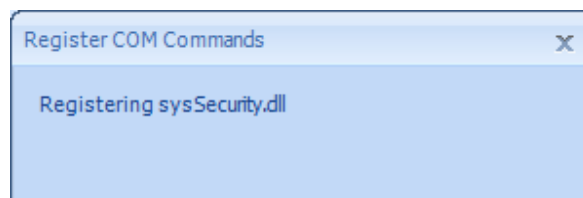
**Setup Options tab**

This will bring up the **Setup** dialog box. Select the **Setup** side tab and the **Get Security Code** button (Register (Overlord1) Commands button).



**Register (Overlord1) Commands button**

A dialog box will be displayed showing the progress of the command registration (Registering (Overlord1) Commands):



**Registering (Overlord1) Commands**

When the window disappears, the process is complete.

Sometimes this registration process can be unsuccessful and the instrument driver of interest does not appear in the instrument list. When this happens please follow this process:

There is a utility in the folder **C:\Overlord\Setup\Register DLLs.exe**. Running this program will register every Overlord1 Driver in this folder.

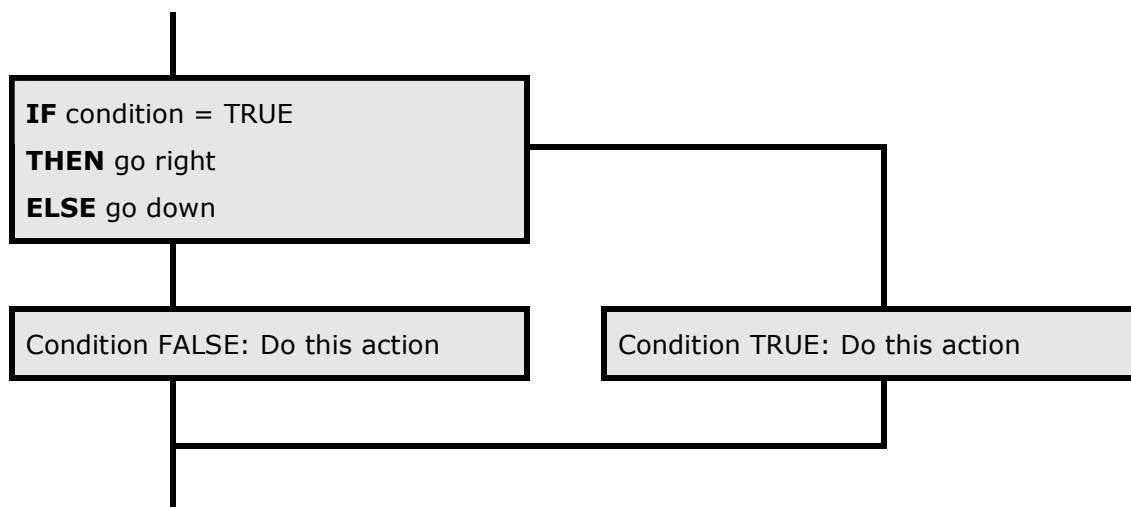


## 4 Commands

All items that are listed in the Command Palette are known as **Commands**. Each command can be dragged onto the **Procedure** flowchart to generate an **Action** which uses the Command in a particular way. The follow section describes all of the common commands available and how they would be used within a **Procedure**.

### 4.1 *Decision*

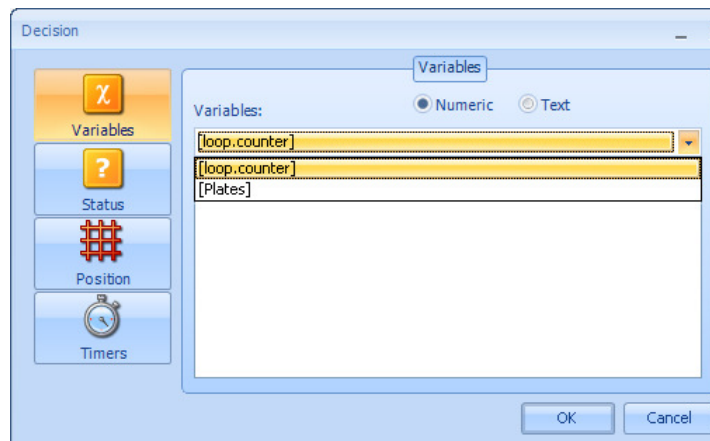
Making a decision allows the user to compare the state of one variable against another.



#### 4.1.1.1 *Variables*

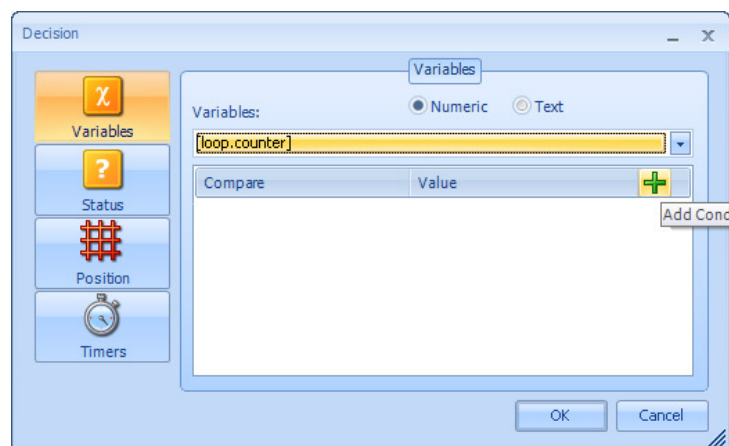
The default on the left of the Decision dialog box is **Variables**. Different types of decision can be achieved with the action (comparison of numbers, comparison of text and determination if a True/False variable is true or false).

The Decision dialog box is:



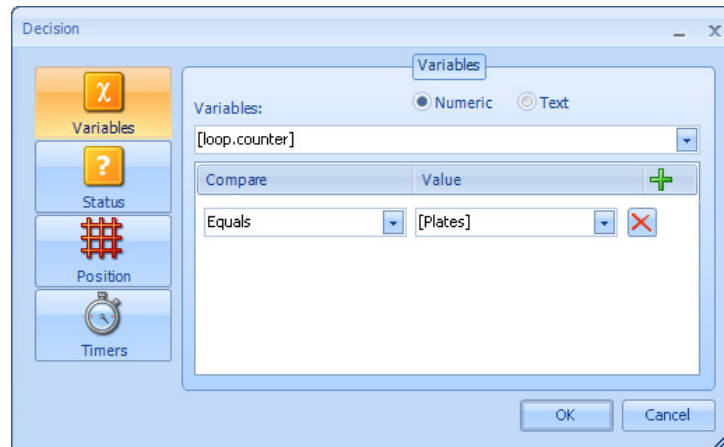
**Decision Dialog Box**

For example if the condition is **[loop counter] = 1**, the variable to make the decision **[loop counter]** is selected from the drop down list. Then the **+** is selected to add the condition.



**Adding a condition**

This displays the condition option box:



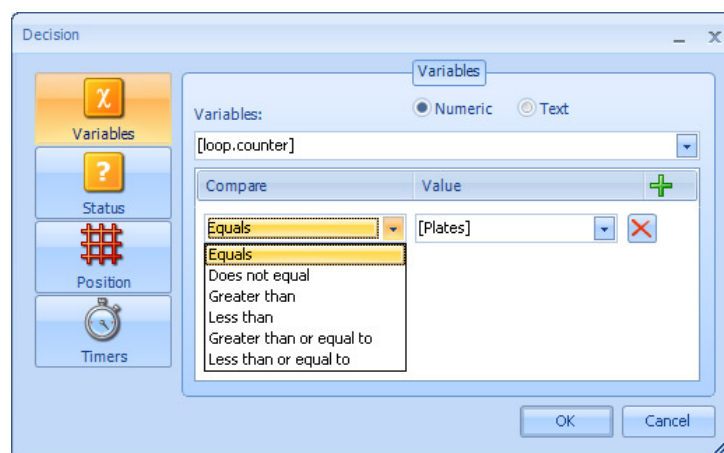
**Displaying the condition options**

The **Compare** type can be selected from:

- Equals
- Does not equal
- Greater than
- Less than
- Greater than or equal to
- Less than or equal to

Text and Yes/No variables can only be **Equals** or **Does not equal**.

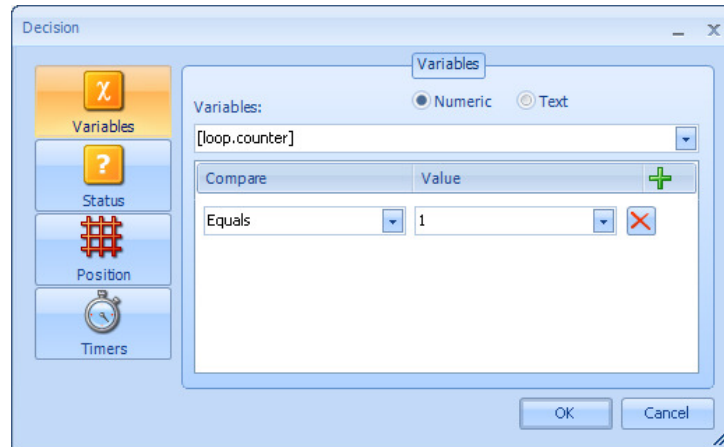
These can be seen below:



**Choosing the Compare type**

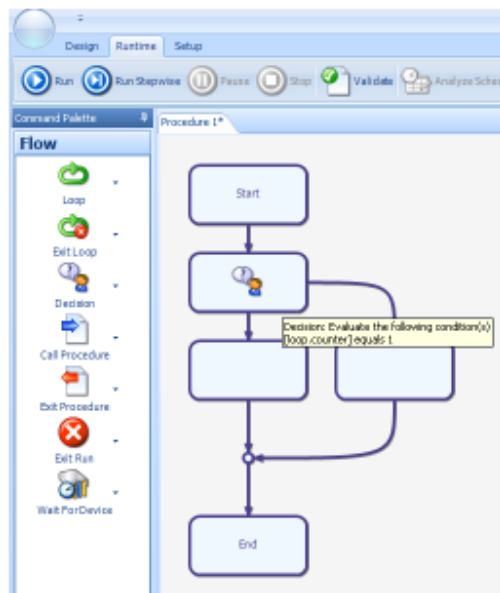
In the example **[loop counter] = 1**, the **Equals** option should be chosen. Finally, the Compare Value should be entered; this can be a variable or a value. For the example, **1** should be entered.

Finally, the Compare Value should be entered; this can be a variable or a value. For the example, **1** should be entered.



**Enter the Compare Value**

This option can be deleted by selecting the red cross at the end of the Compare line. Selecting the **OK** button, will give the following procedure steps:

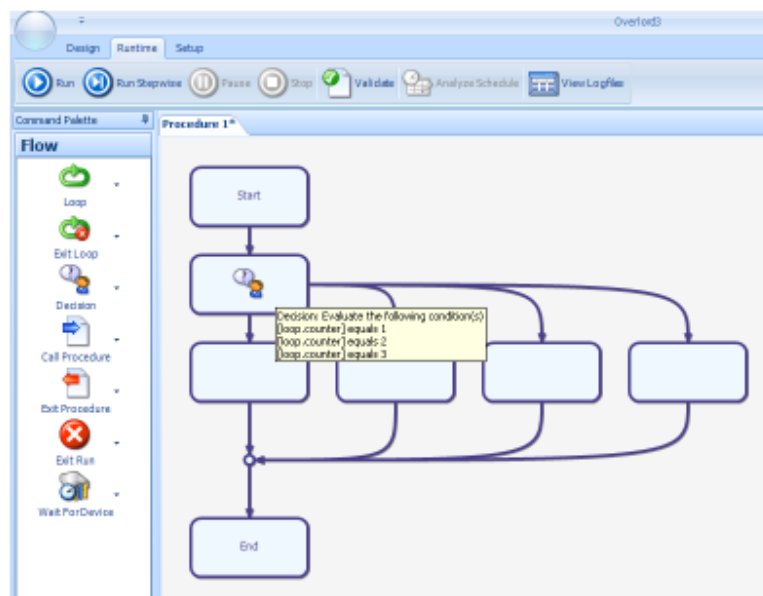


**The final procedure for this Decision step**

If the **Decision** step is **TRUE** that is **[loop variable] = 1**, the right hand path will be run. If the **Decision** is **FALSE**, the left hand path will be run.

#### 4.1.1.2 Advanced

This statement can also be used to run multiple options, sometimes known as a SELECT CASE statement. If the user wishes to test if **[loop variable] = 1, 2 or 3**, and complete different operations based on the value of **[loop counter]**. The user should extend the Decision dialog with an additional couple of steps:

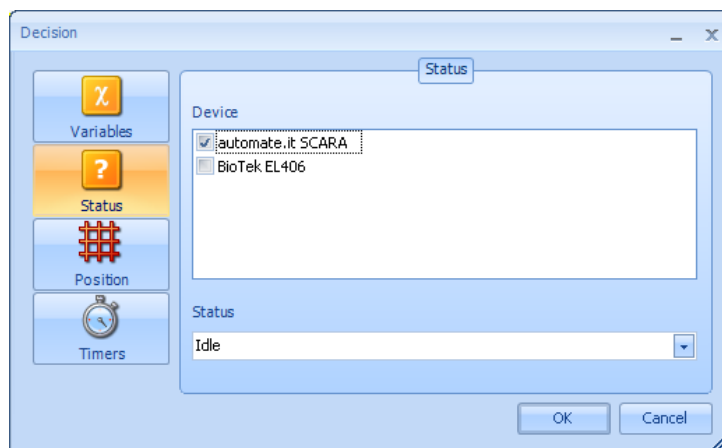


#### Multiple options in a Decision procedure step

If **[loop counter]** is not equal to 1, 2 or 3, the left hand path will be run. If **[loop counter]** is equal to 1 the second path from the left will be run. If **[loop counter]** is equal to 2 the third path from the left will be run. If **[loop counter]** is equal to 3 the right hand path will be run.

## 4.1.1.3 Status

Selecting the **Status** option on the left of the **Decision** dialog box, changes the options. Status refers to the runtime state of the instruments installed on the system. In this example, an automate.it SCARA and a BioTek EL406.

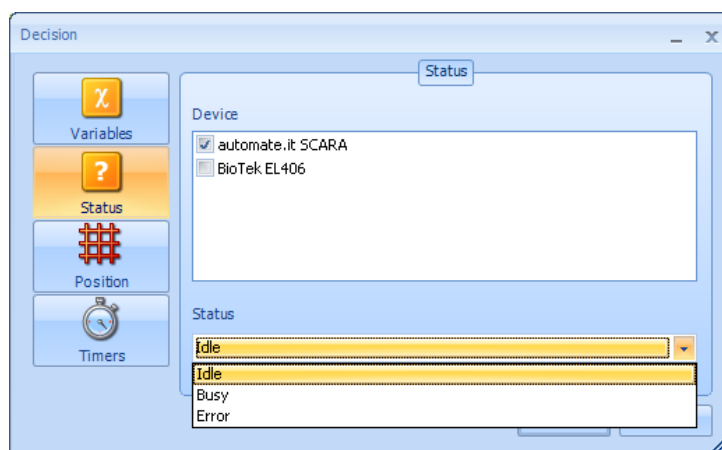


**Status options**

The instrument activity can be in one of three states:

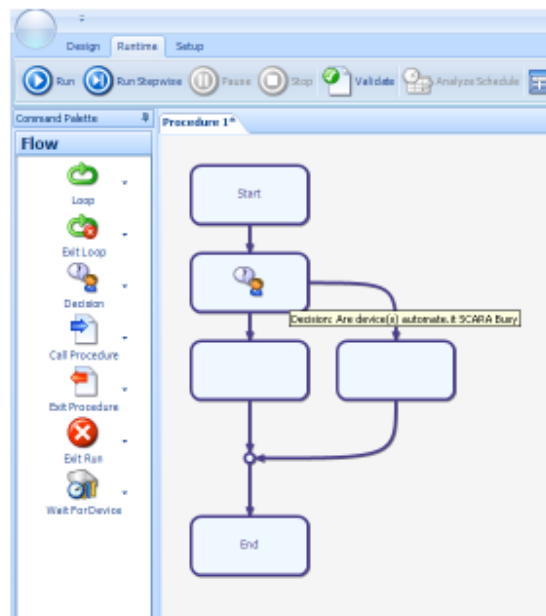
- Idle
- Busy
- Error

These options are displayed as below:



**Instrument states**

The resulting procedure is:

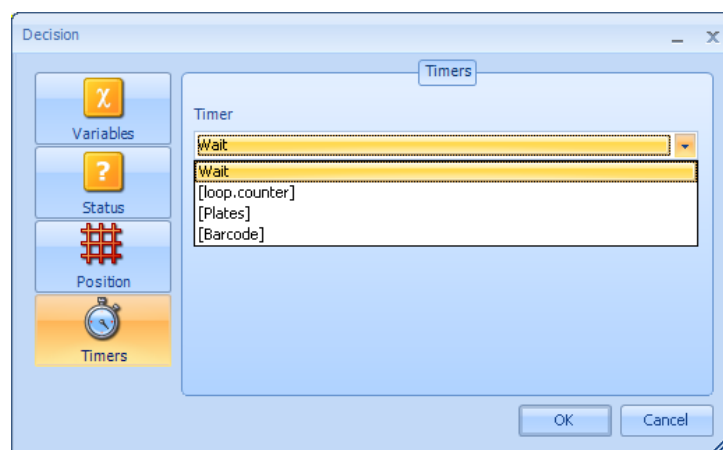


### Checking the status of an instrument

If the robot is **Busy**, the right hand path will be followed. If the robot is **not Busy** the left hand path will be followed.

#### 4.1.1.4 Timers

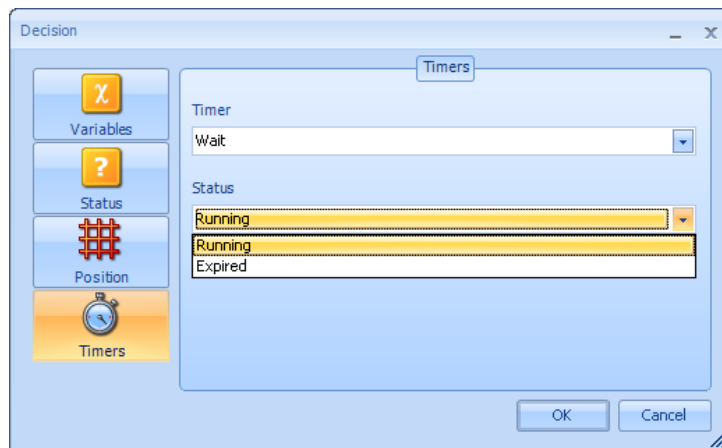
Selecting the **Timers** option on the left of the **Decision** dialog box, changes the options. Status refers to the runtime state of the timers used on the system. In this example, there is the system Wait timer (see Wait Timer) and the other variables that may represent a timer.



**The Timers Decision Dialog Box**

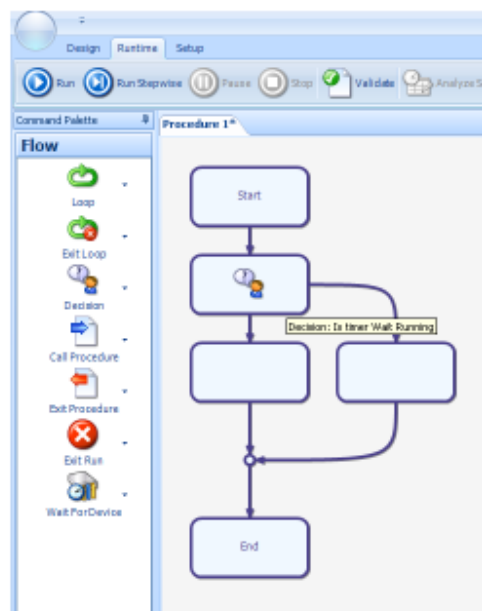
Choose the Timer that you wish to check, then select the state of the timer in the lower drop down box, the options are:

- Running
- Expired (complete the set time)



## Timer state selection

Selection of the **OK** button, give the following procedure steps:



## Decision step for Timers option

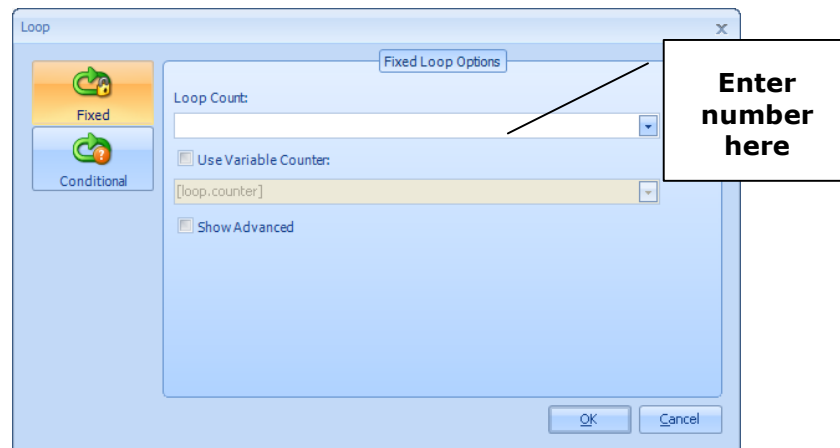
If the chosen timer is **Running** the right hand path will be run. If the Timer is **not Running**, the left hand path will be run.



## 4.2 Loop

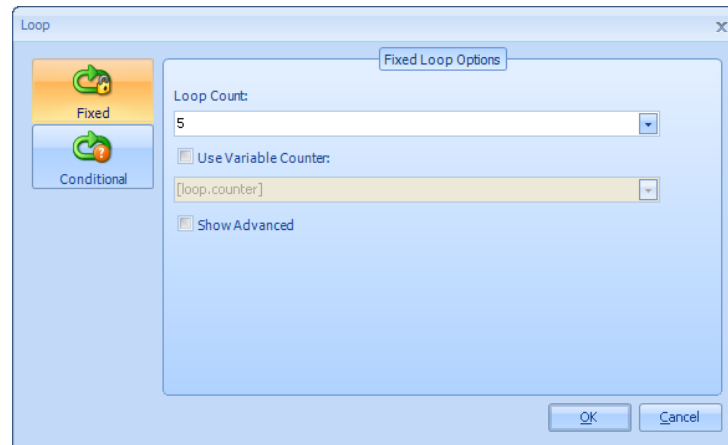
### 4.2.1 Running a simple loop

The **Loop** command allows the user to construct a procedure that can repeat a number of actions again and again. Drag a **Loop** command from the **Flow palette** to the procedure and the **Loop** dialog box is displayed. Keep the **Fixed** option selected.



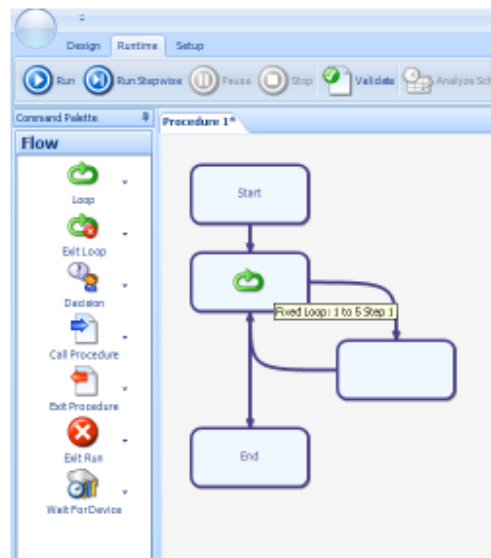
**Initial Loop Dialog box**

If you enter a number into the **Loop Count** text box:



**Enter a number for the Loop Count**

Selecting **OK** will give the following procedure:

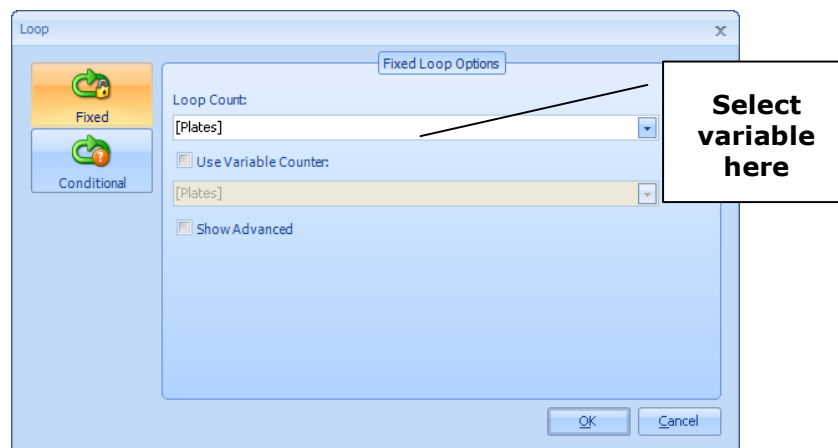


### Result of entering a Loop Count of 5

This action will get the procedure to loop around running any actions inside it 5 times

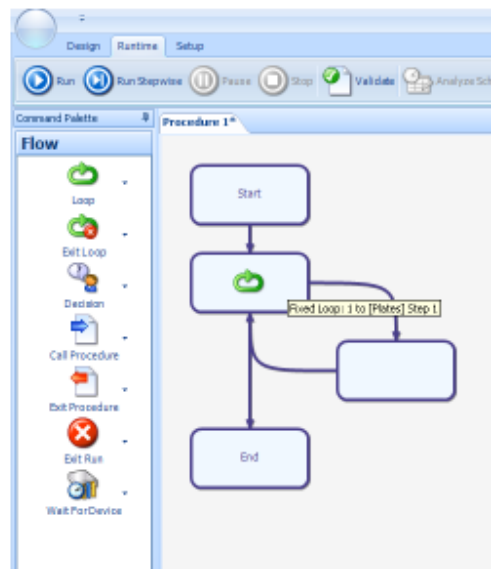
#### ***4.2.1.1 Using a variable to control the number of loops***

Entering a variable value into the procedure step (double left hand click on the **Loop** icon) gives:



### Changing to a variable loop

Selecting OK gives the same flow chart as above, but with a different runtime profile.

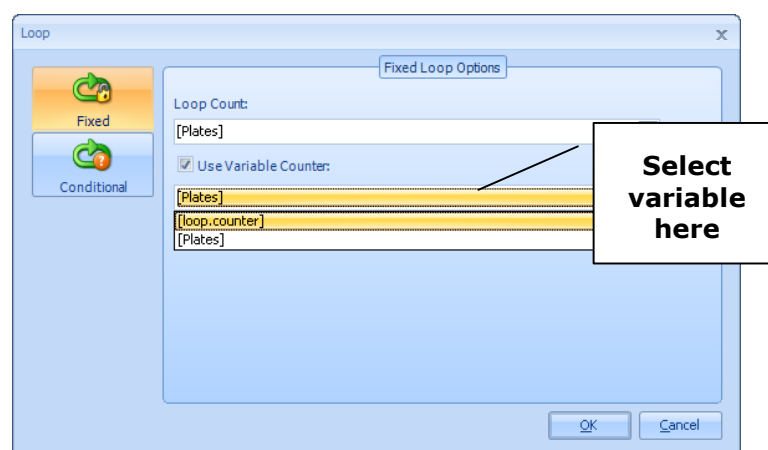


### Result of entering a variable Loop Count of [loop counter]

This will run the loop a variable number of times determined by the value of [loop counter] at that point at runtime. So if [loop counter] is equal to 4, the loop will run 4 times; if [loop counter] is equal to 20, the loop will run 20 times.

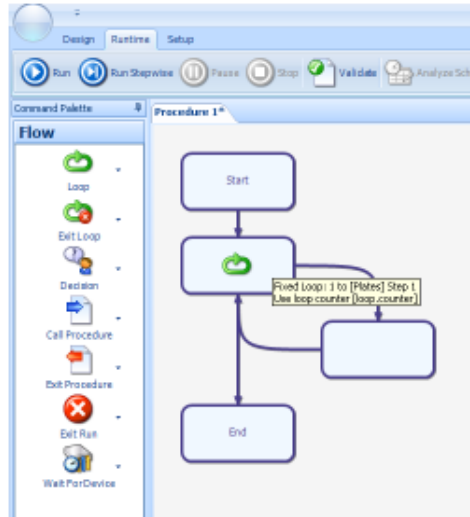
#### 4.2.1.2 Accessing the looping variable

It is possible to know what the current loop number is. Click the **Use Variable Counter** check box and select the counting variable name from the drop down list. In this example **[loop counter]** is selected.



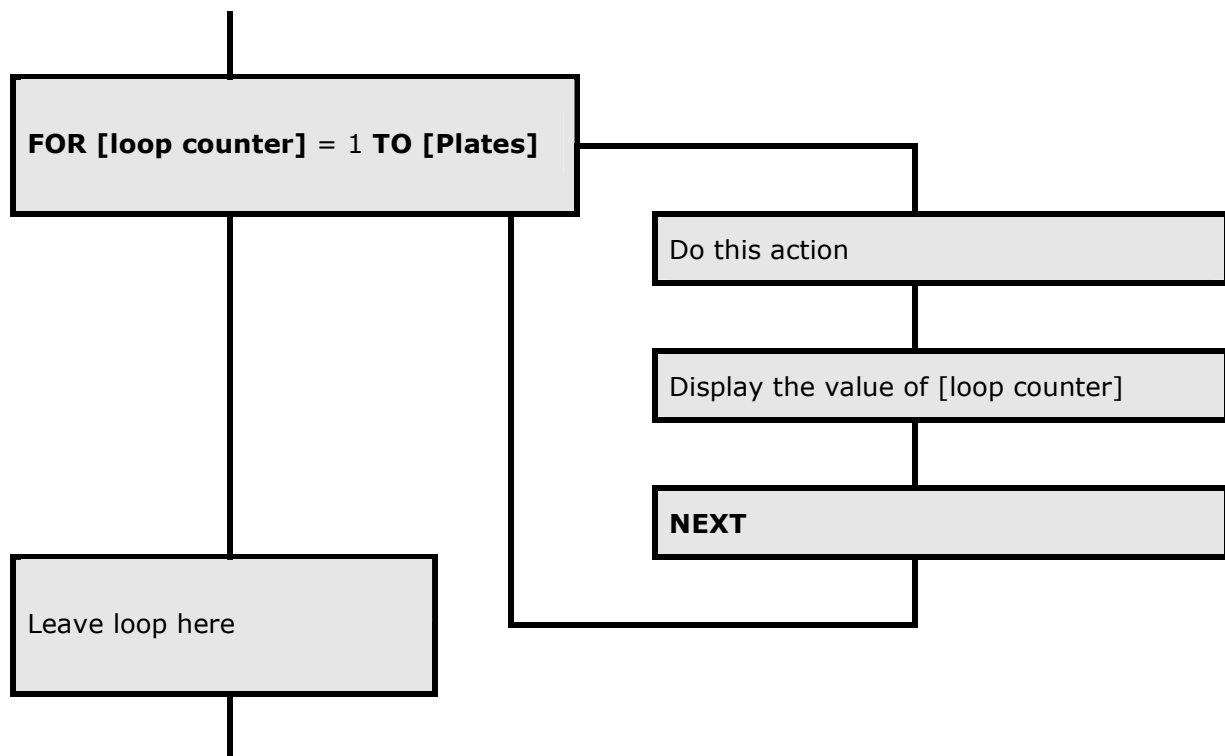
### Loop with named looping variable

This gives the following flow chart:



## Loop with accessible counting variable

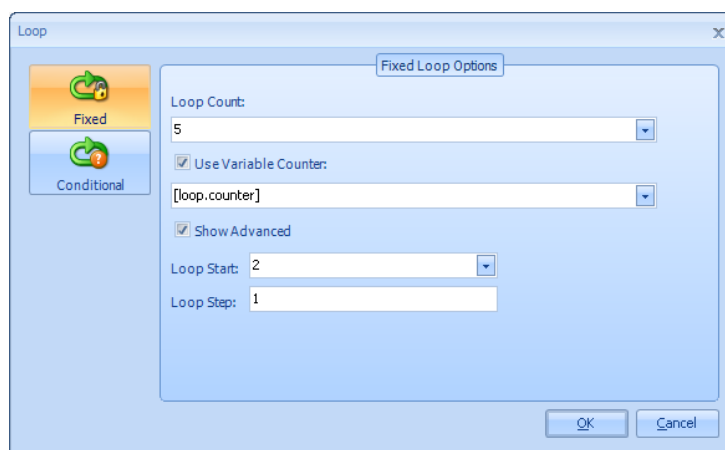
This will allow the user to use the counting variable inside the loop as shown below:



An example would be if **[Plates]** was set to 5 (see setting variable values), then the value of **[loop counter]** would be first one, then 2, then 3, then 4 and finally 5. The procedure would then continue out of the bottom of the **Loop** step.

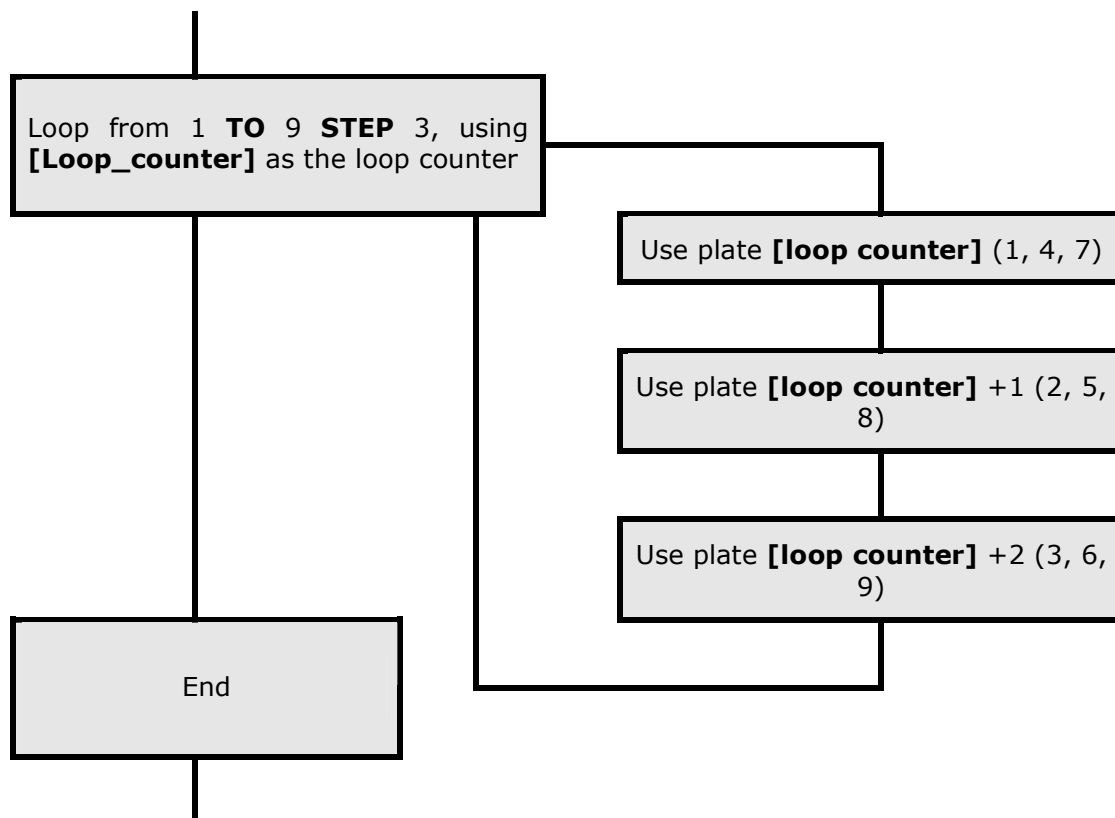
#### 4.2.1.3 Changing the starting value for the loop and the step

In each of the above examples the loop counter starting position and the step value has a default value of 1. This is normal for most examples. However, if the user needs to start from a different value other than one, for example start at a value 2 and finish at 5, the **Advanced** option needs to be selected.

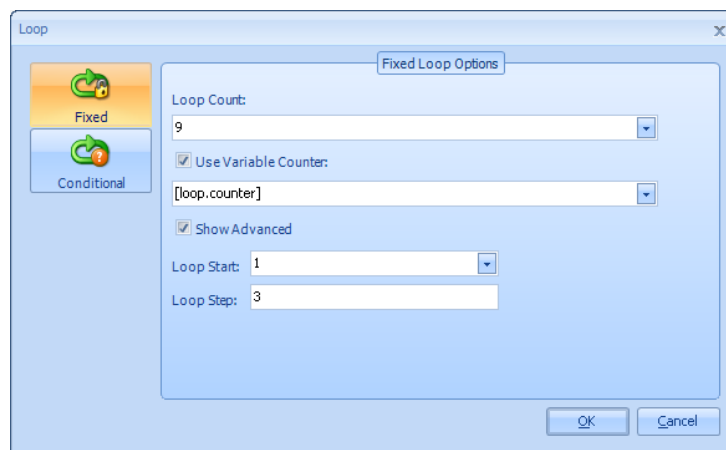


**Making the Loop run from a value of 2 to 5**

In addition, in very complex cases, the user may wish to go from 1 to 9 in jumps of 3. An example of this would be when plates are removed from a stack 3 at a time. So the first pass of the loop will use plates 1, 2 and 3; the second pass of the loop would use plates 4, 5 and 6, etc. This is shown diagrammatically below:

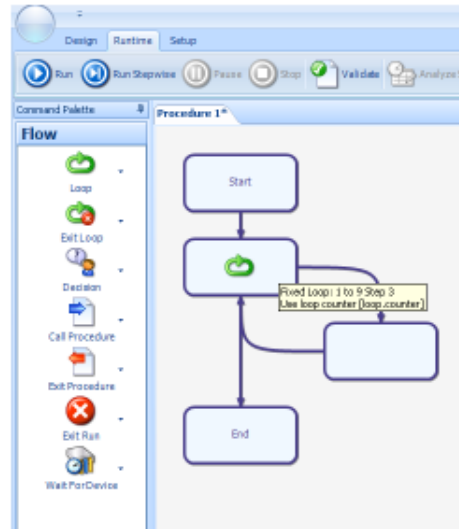


The Loop flow chart procedure step would be modified as below:



**Loop from 1 to 9 step 3**

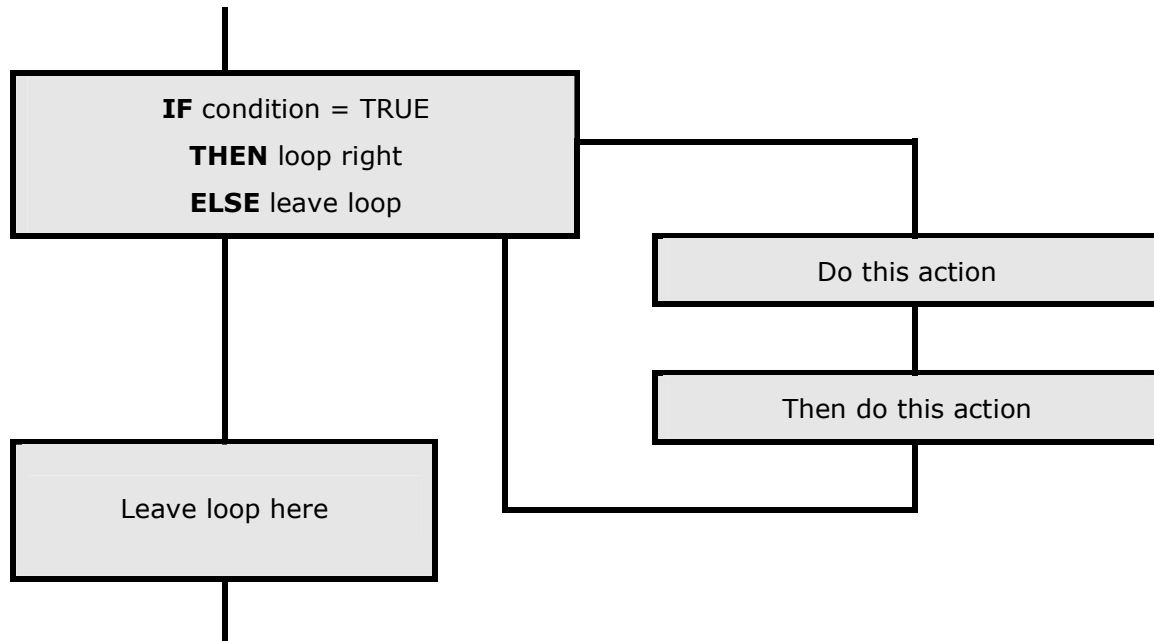
The procedure would then look like this:



**Procedure with a stepped loop**

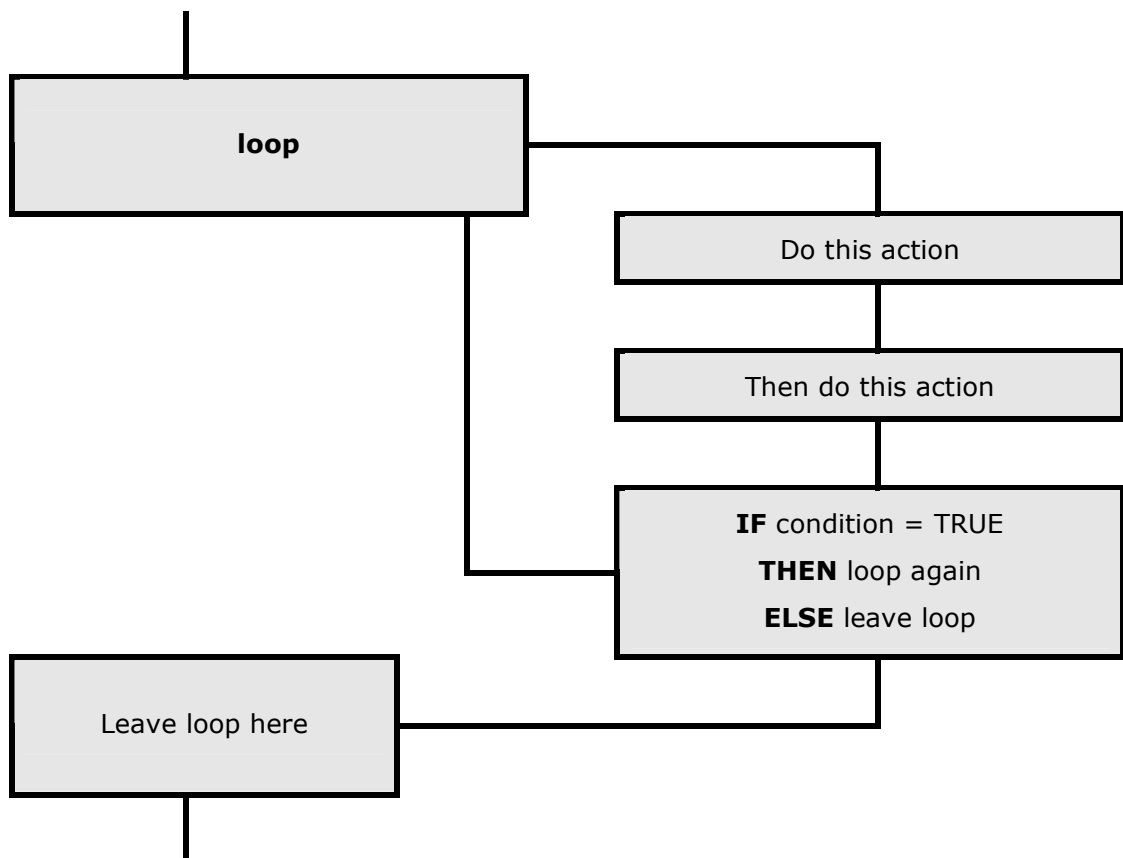
### 4.2.2 Looping on a condition

Looping on a condition allows the user to complete a loop a set number times, but the decision is based on a condition at runtime. This conditional test can be completed at the beginning of a loop:



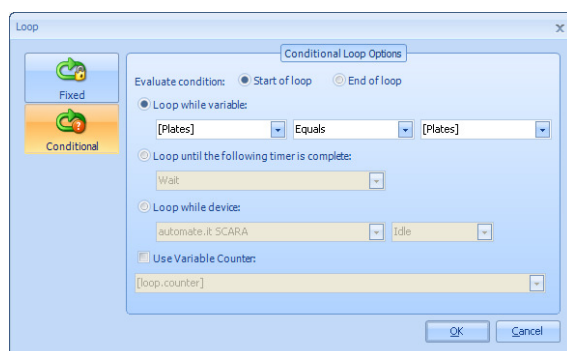


Or at the end of the loop:

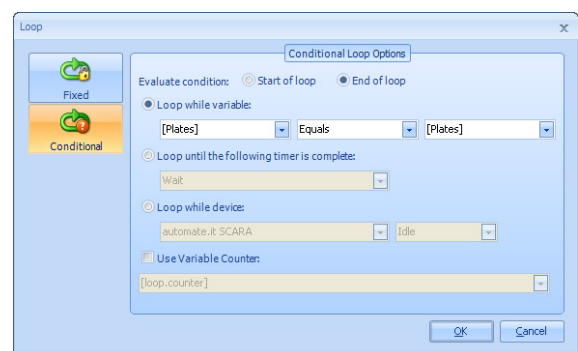


The key difference is that the test at the top of the loop, the loop may never run; if the test is completed at the end of the loop, that loop will run at least once.

Each of the following can be applied to testing at the start or the end of the loop.



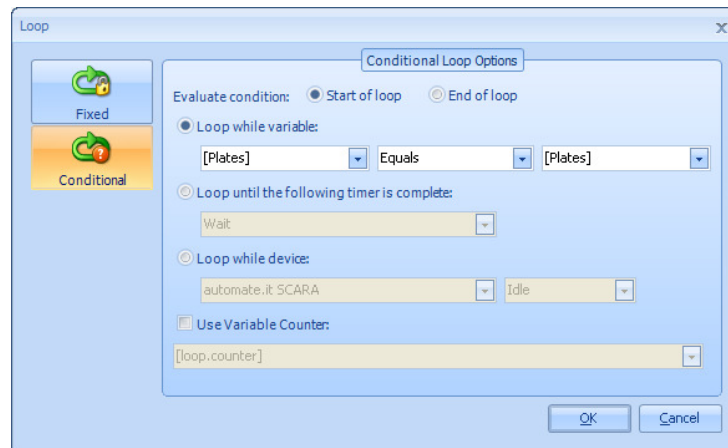
**Test at Start of loop**



**Test at End of loop**

#### 4.2.2.1 Testing at the top of the loop with "Loop while variable"

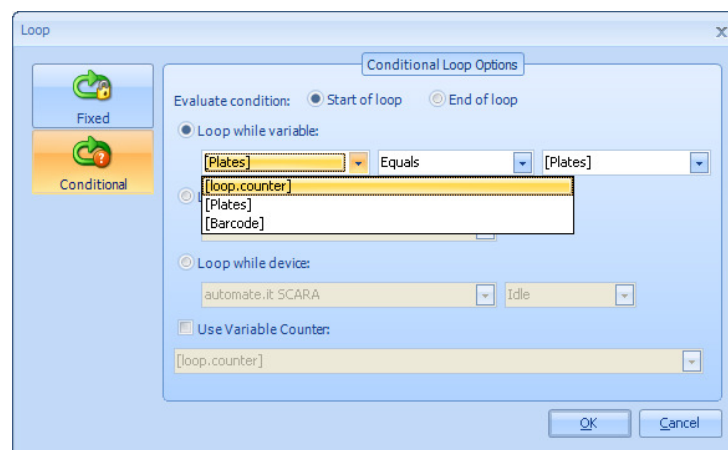
Drop a Loop step into the procedure and select the **Conditional** option on the left of the dialog box, and select the **Start of loop** option button.



**Select conditional at the start of the loop**

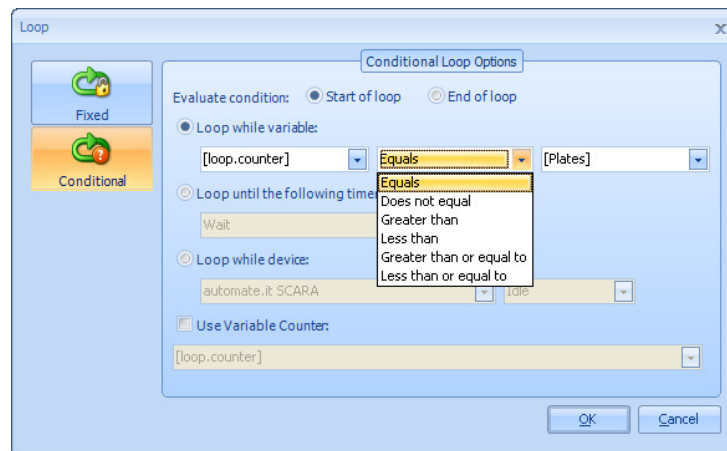
The loop conditions are set under the **Loop while variable**.

Choose the key looping variable:



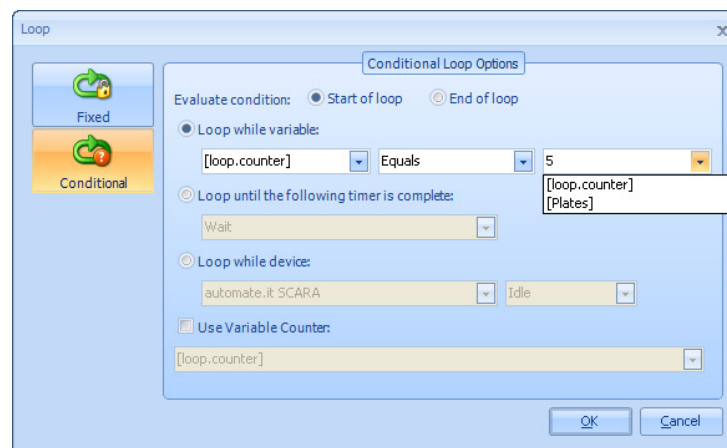
**Select the required looping variable**

Then the inequality from the list:



**Select the required inequality**

Then finally the comparison, this can be another variable or number/text value:

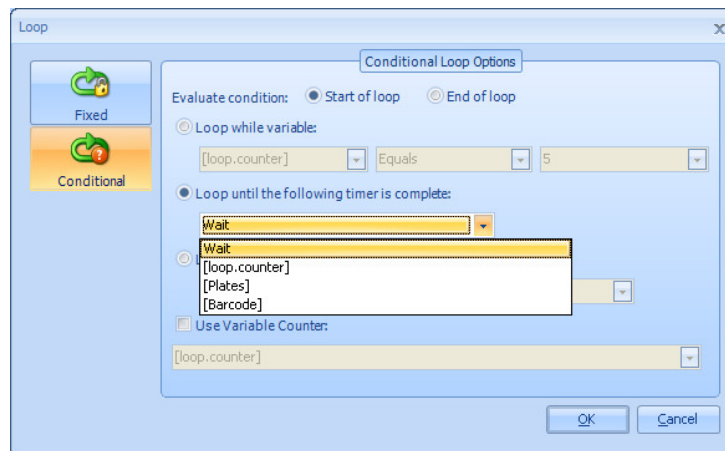


**Set the comparison variable**

## ***4.2.2.2 Testing at the top of the loop with "Loop until the following timer is complete"***

This option allows a timer to be checked to see if it has reached its target time. The user can either choose the wait timer or any of the variables. See Section 0

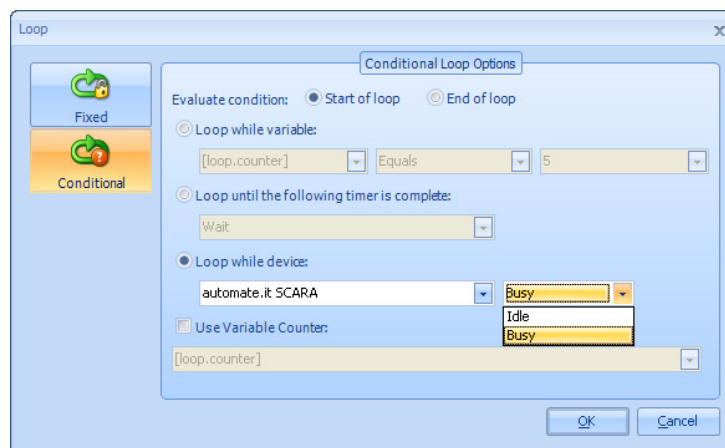
Timers for more information on Timers.



Select the timer comparison option

### 4.2.2.3 Testing at the top of the loop with "Loop while device"

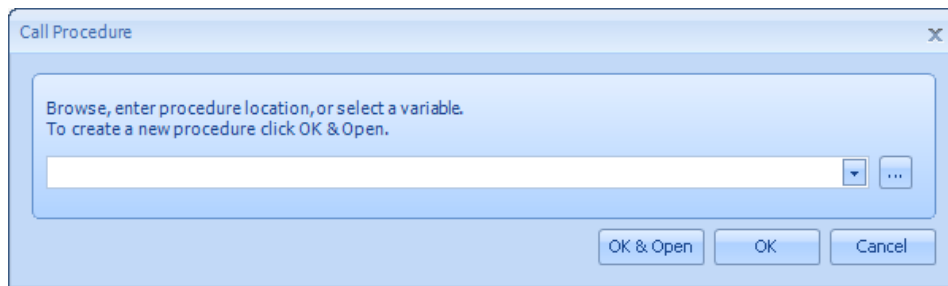
This allows the user to test if one of the instruments is **Busy** or **Idle**. The drop down list is filled with the instruments installed on the system.



Select the Loop while Device option

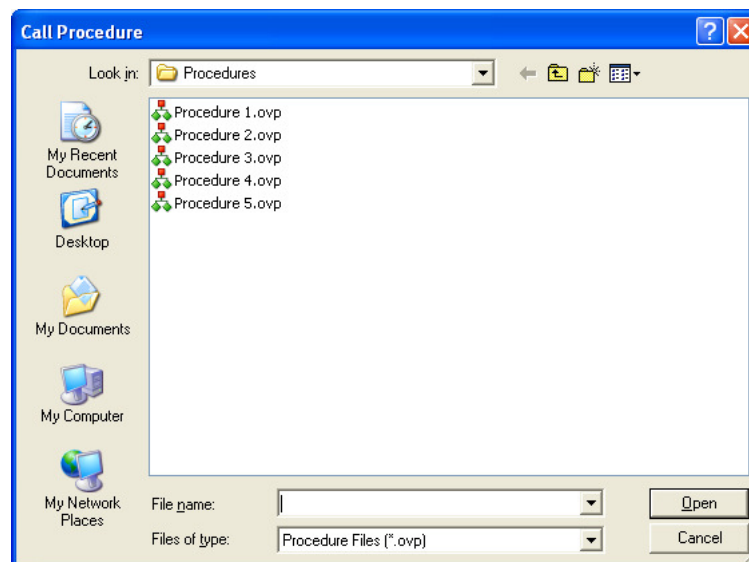
## 4.3 Call Procedure

The **Call Procedure** step allows the user to call a previously written and saved procedure. On dropping on the procedure, the following dialog box will appear:



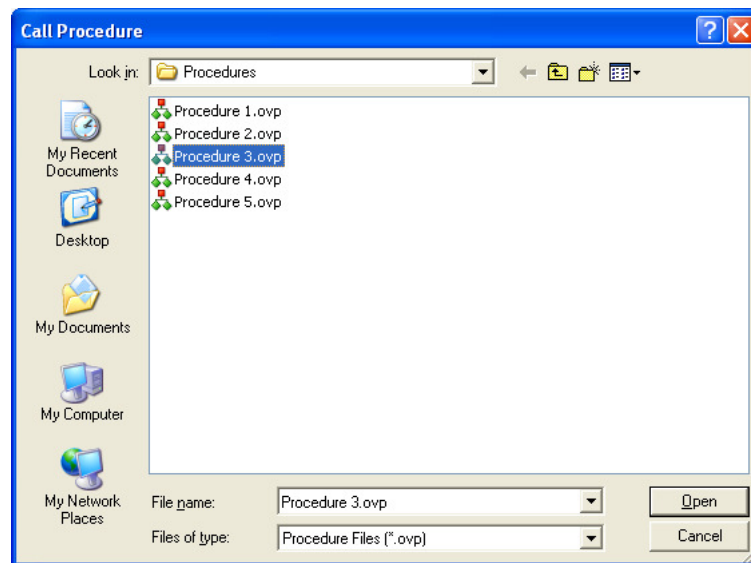
**Call Procedure Dialog Box**

It is possible to type the required file name into the text box, however, it is easy to make typographical mistakes and it is recommended that the user select the required procedure name from the list. Click the ... icon to open the File Open dialog so the procedure can be selected.



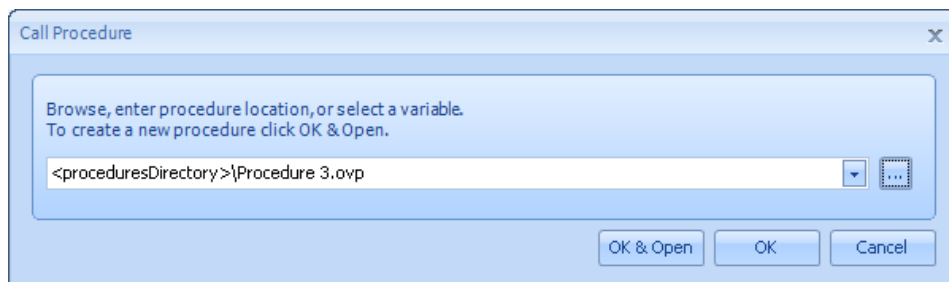
**Call Procedure File List**

Click on the required Procedure file ...



**Call Procedure File List with Procedure File selected**

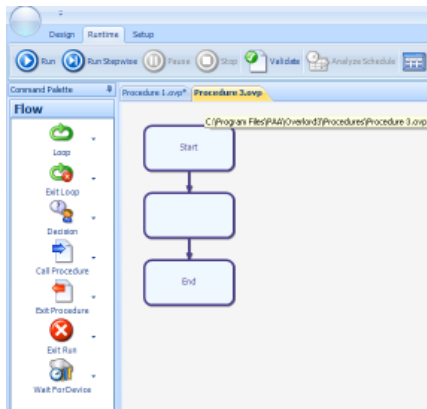
... and the select **Open**.



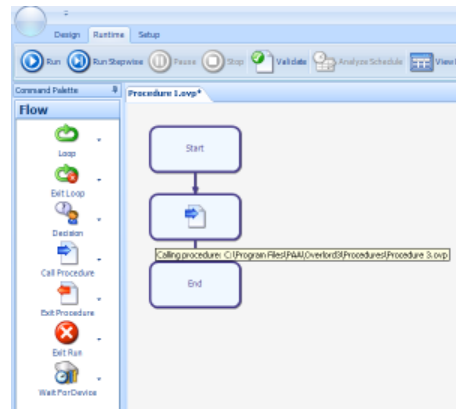
**Call Procedure Dialog Box with file name filled in**

The user can either select **OK**, and that file will be used in the **Call Procedure step** or select **OK and Open** to show both the calling file and the called file.

In this example, the opened file is blank.



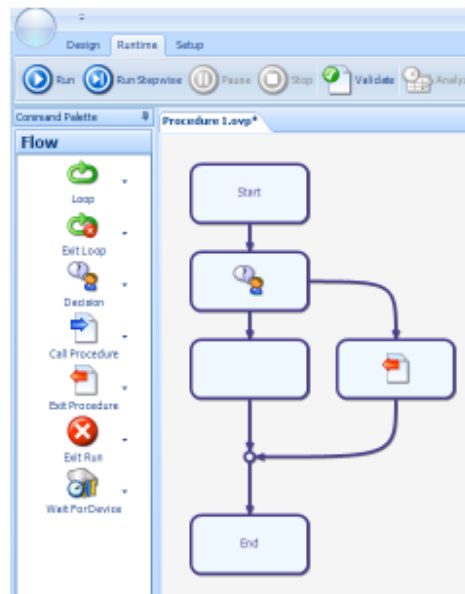
**Select OK and Open**



**Select OK**

## 4.4 *Exit Procedure*

The **Exit Procedure** command allows the user to exit a running procedure when the processing has completed. The **Exit Procedure** command has no edit screen and can be used anywhere in a procedure.



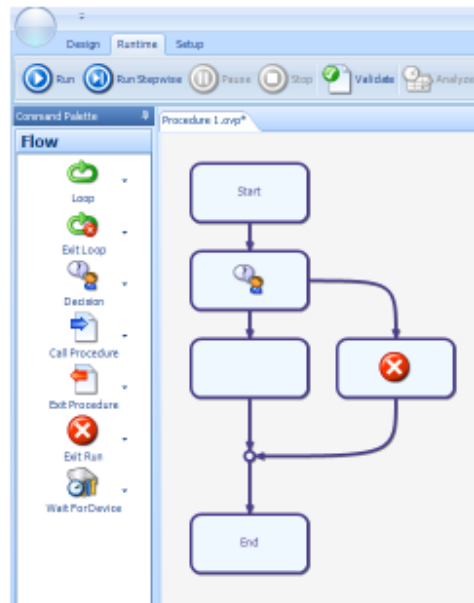
**Exit Procedure step added to the procedure**

In this example, if the decision gives a value of TRUE, the procedure will exit to the calling procedure.

## 4.5 Exit Run

The **Exit Run** command allows the user to exit a running procedure ending the run completely.

The **Exit Run** command has no edit screen and can be used anywhere in a procedure.



### Exit Run procedure entered in the procedure

In this example, if the decision gives a value of TRUE, the procedure will exit the run completely.

## 4.6 Initialize Variables

The **Initialize Variables** command allows the user to initialize all the declared variables. The declared variables are found in the variables window, usually in the top right hand side of the screen.

Variables			
<div> <span>+</span> <span>×</span> <span>↑</span> <span>↓</span> <span>Name</span> <span>↑</span> <span>↓</span> <span>Type</span> <span>✓</span> <span>Update</span> </div>			
Name:			
Value	Name	Type	
abc	[Barcode]	Text	
38	[loop.counter]	Numeric	
12	[Plates]	Numeric	
False	[ready]	TrueFalse	

**Set and View Variables window**



If the initialize command is used in the procedure as below:



**Initialize Variables procedure step**

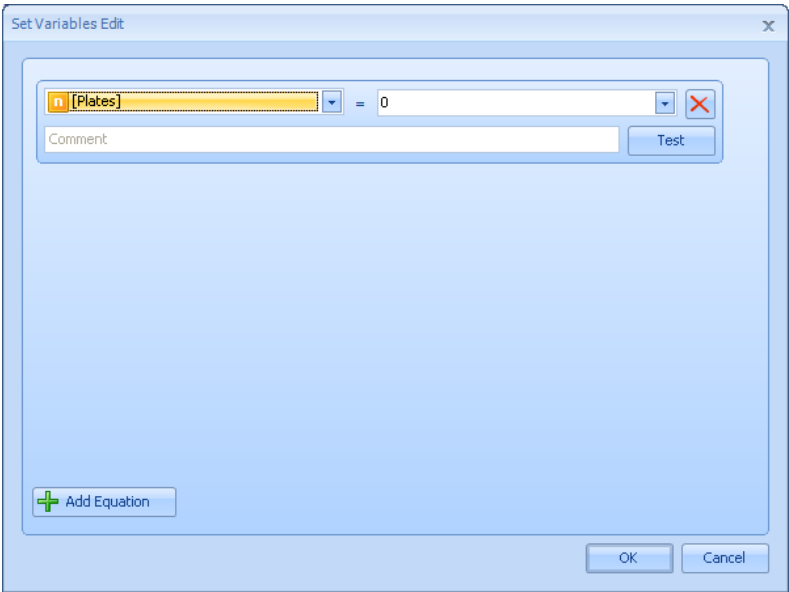
When this procedure is run, the variables are initialized.

Variables			
<div> <span>+</span> <span>×</span> <span>↑</span> Name                     <span>↑</span> Type                     <span>✓</span> Update                 </div>			
Name: [loop.counter]			
0			
Value	Name	Type	
	[Barcode]	Text	
0	[loop.counter]	Numeric	
0	[Plates]	Numeric	
False	[ready]	TrueFalse	

**Variables Window shows that the variables have been initialized**

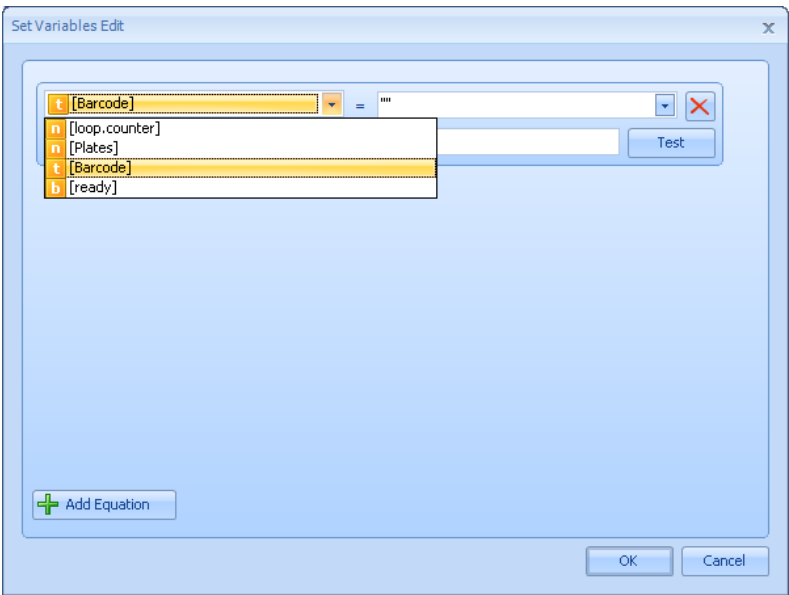
## 4.7 Set Variables

The **Set Variable** command allows the user to set a variable to the required value. When this command is placed in the procedure, the edit screen is displayed.



**Choose the variable to be set**



The user can then choose the variable to be modified:

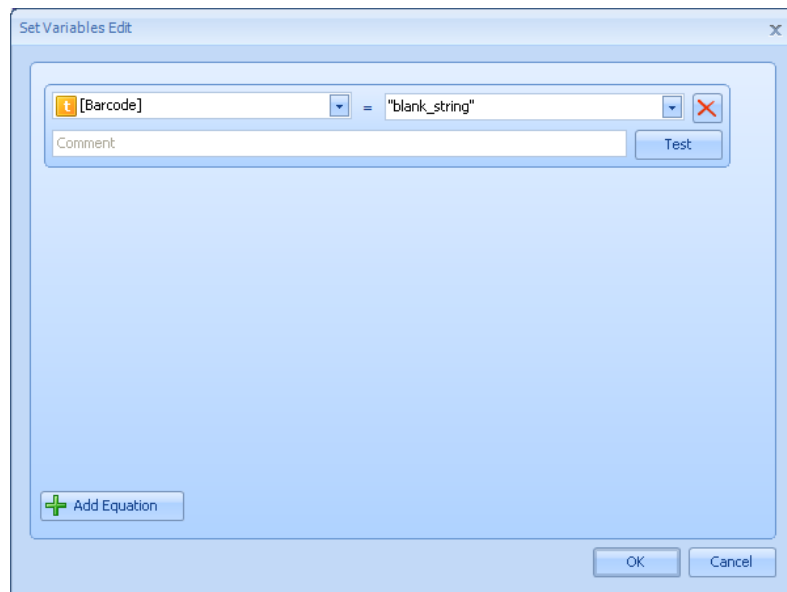


**Select the variable name from the drop down list**

The orange box indicates the variable type.

	Numeric	Any number
---	---------	------------

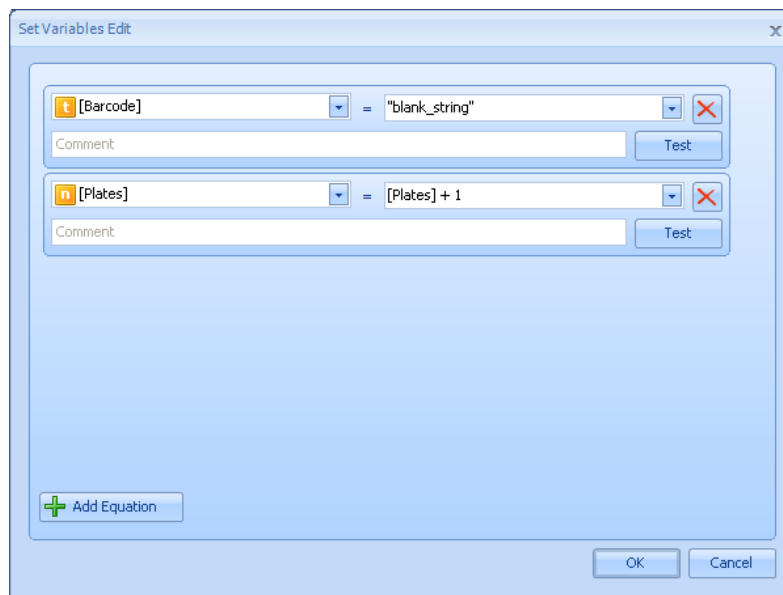
	Text	Any non-numeric piece of data
	True/False	Choice of TRUE or FALSE



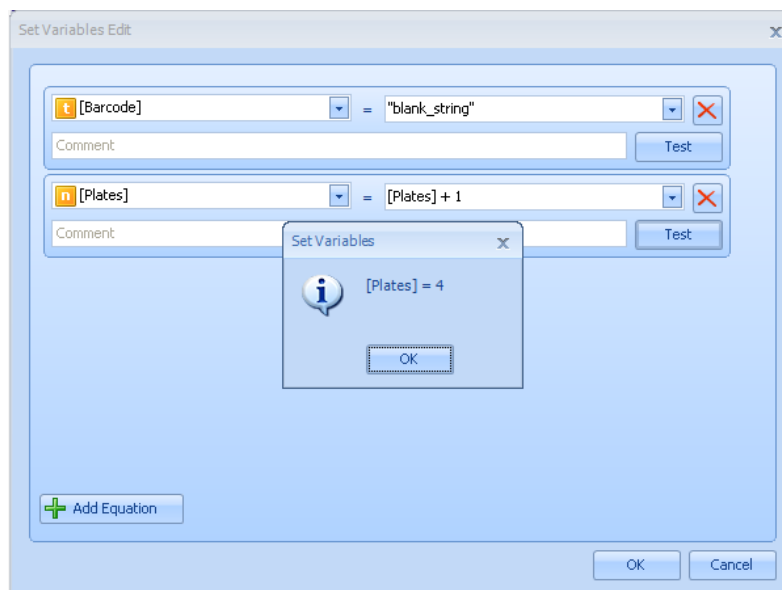
**Enter the value to be assigned to the variable name**

The choice of number/text/yes-no will affect the acceptable entries for the variable.

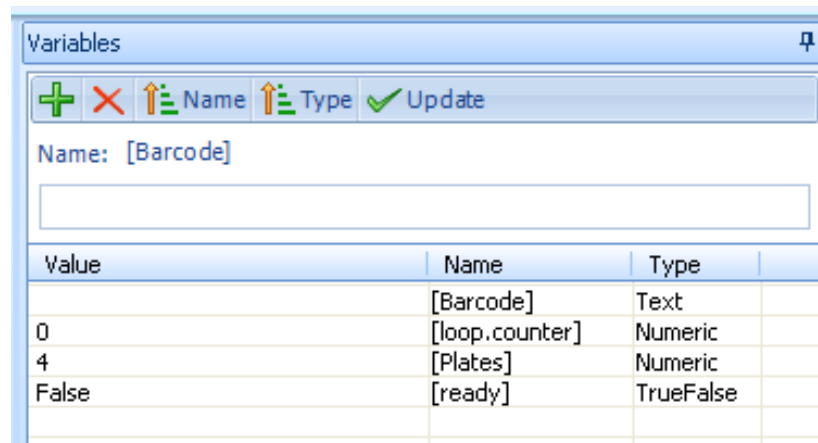
Multiple variables can be assigned in a single procedure step. Use the **Add Equation** button for each additional assignment.

**Add an additional set variable line**

Equations can be entered into the value box. The assignment can be tested. For example, in the example above if [plates] = 3, selecting the Test button will display the new value.

**Testing the Set Variable option that has been entered**

Select OK to acknowledge the new value, which will be visible in the variables window.

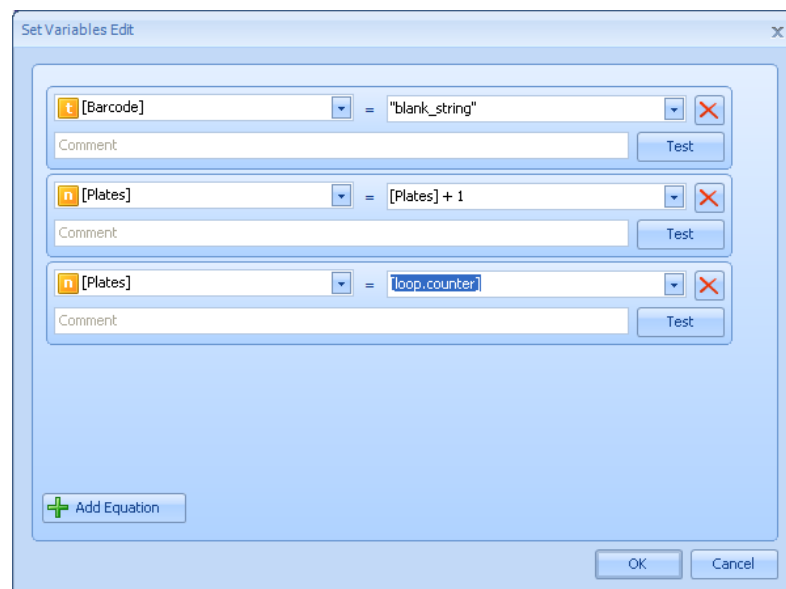


The Variables window displays a table of variables. At the top, there are icons for adding (+), deleting (X), and sorting (arrows) by Name, Type, and Update. Below these is a search bar labeled 'Name: [Barcode]'. The main table has columns for Value, Name, and Type.

Value	Name	Type
	[Barcode]	Text
0	[loop.counter]	Numeric
4	[Plates]	Numeric
False	[ready]	TrueFalse

**Variable window after the Set Variable step has been run**

Assignment of one variable to another variable is also possible.



The Set Variables Edit dialog box allows for editing variable assignments. It contains three rows, each with a variable name in a dropdown, an equals sign, and a value in another dropdown. Each row has a 'Test' button and a 'Comment' text area. At the bottom, there is an 'Add Equation' button and 'OK' and 'Cancel' buttons.

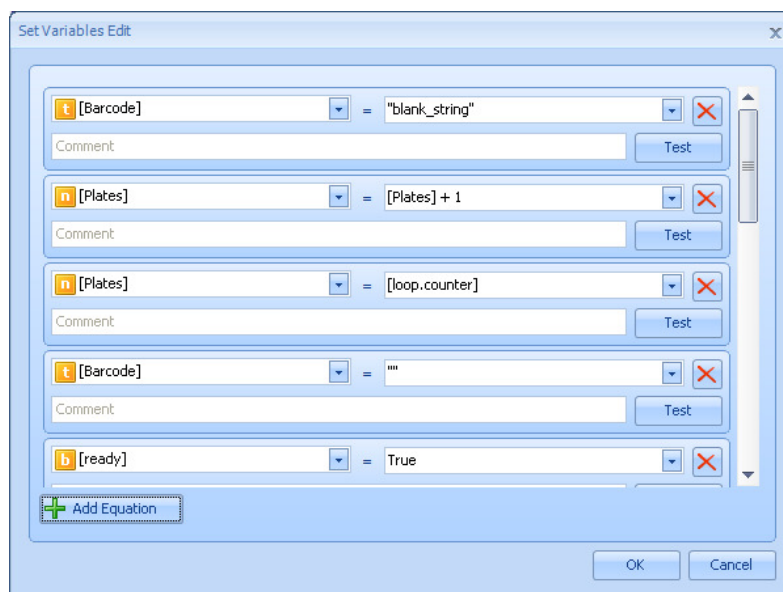
Variable	Assignment
[Barcode]	"blank_string"
[Plates]	[Plates] + 1
[Plates]	[loop.counter]

**Assigning variable to another variable name**

Any of the assignments can be deleted with the red cross.



Additional assignments will lead to the display of a scroll bar.



**Result after multiple Add Equation clicks**

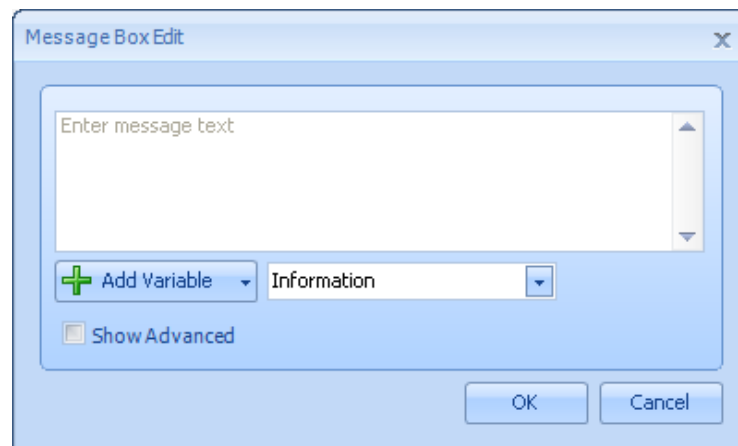
Selecting the OK button will display this command in the procedure.



**The values of the Set Variables are visible in the flow chart**

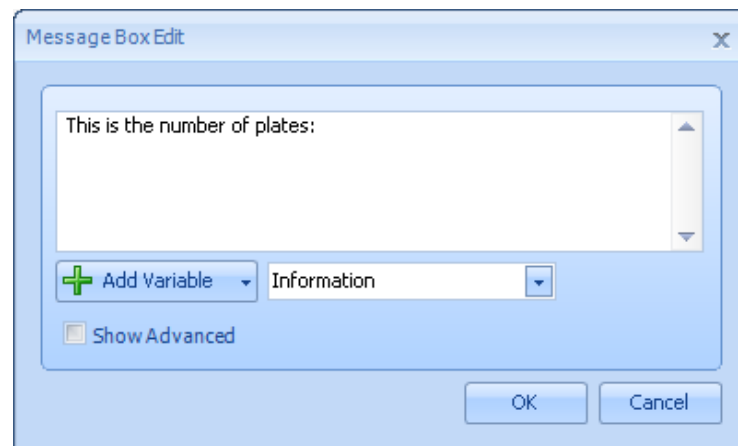
## 4.8 Message Box

The **Message Box** command is used by the person setting up the system to ask the user of the system to complete a task. Such actions could be "check a pipetting station waste bottle is empty" or "load the stack with microplates". The edit screen is as below:



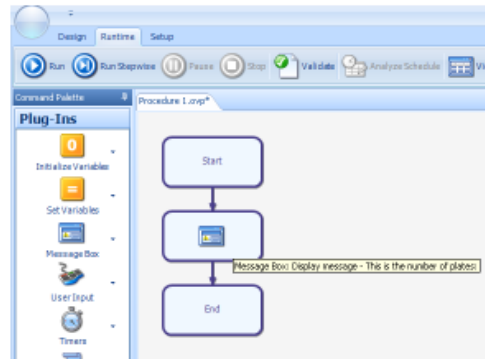
**The Message Box edit box**

The user puts the message into the text area:



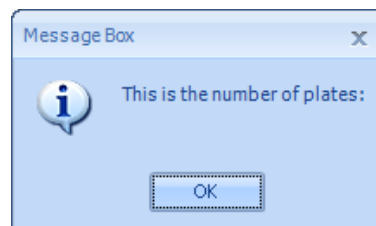
**Enter the message into the text window**

Press **OK** to return to the procedure.



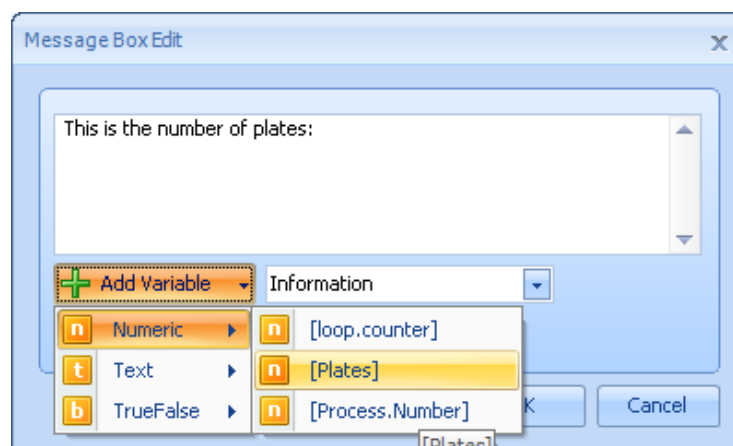
## The message box contents in the tooltip

Save and then run the procedure displays the message:



## The message at runtime

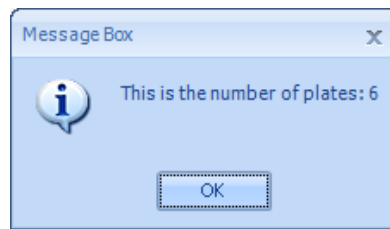
Any of the variables can be incorporated into the message. Place the cursor where you want the variable to be entered, select the variable name from the list:



## Add a variable to the message

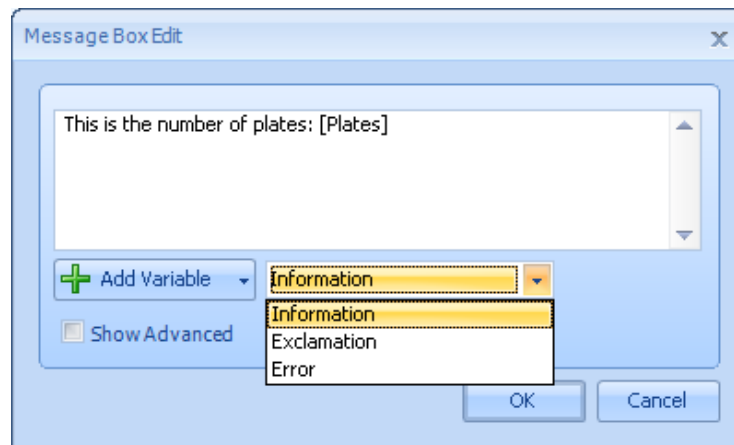
If the procedure step is exited, the procedure saved and run, the value of the variable is displayed:





## Message at runtime

The symbol may be changed with the other drop down list:



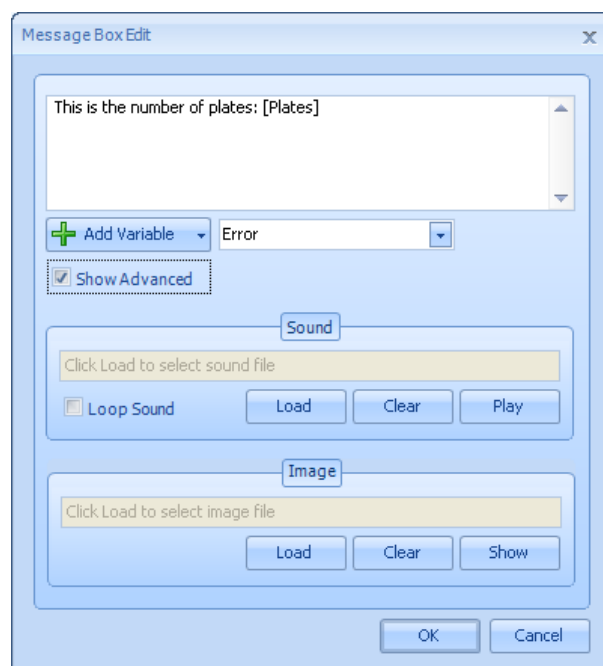
## Changing the symbol in the message window

The symbols are:

<b>Information</b>	
<b>Exclamation</b>	

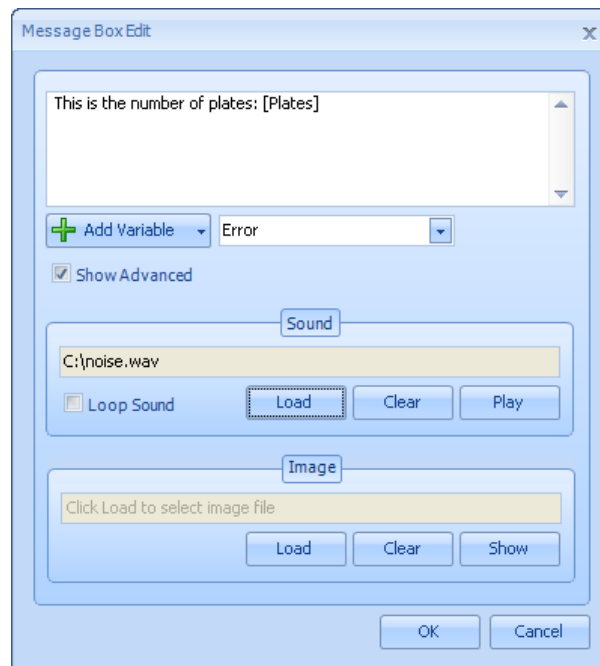
<b>Error</b>	
--------------	--

The final option is the facility to play a wave file at runtime. This is useful to issue "verbal" messages to the user at runtime, in any language. Use Windows Sound Recorder and a microphone to record the required message/sound. Select the **Show Advanced** check box.



## Adding a sound recording to the message

Click on the **Click load to select sound file** and use Load to select the filename of the sound to play.



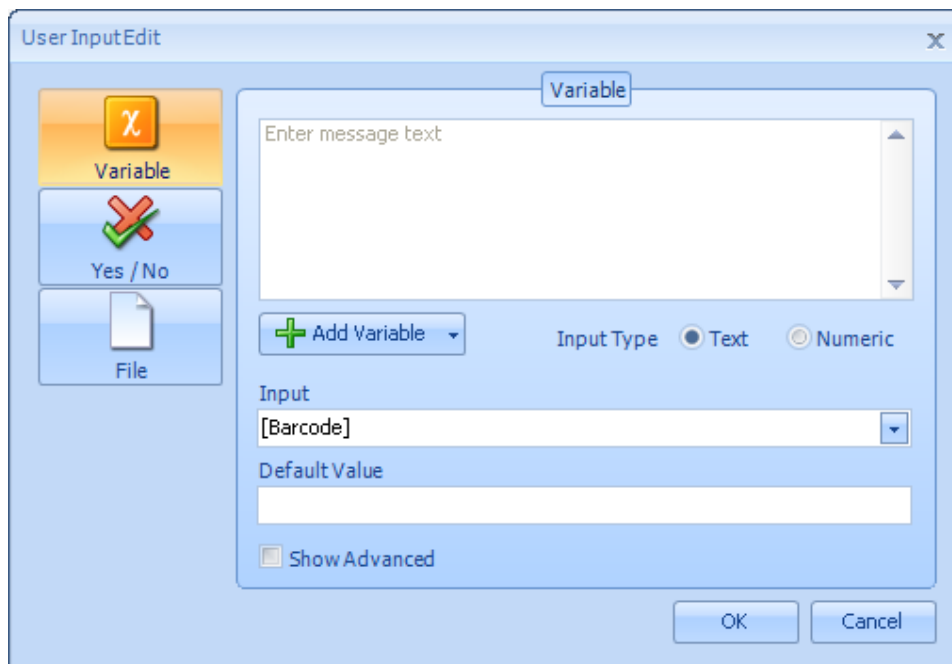
**Loading the sound recording file**

Use **Clear** to delete the current filename and **Play** to test the sound. Select the **Loop Sound** check box to run the sound on a continuous loop. The sound will stop when the **OK** button is selected.

## **4.9**     ***User Input***

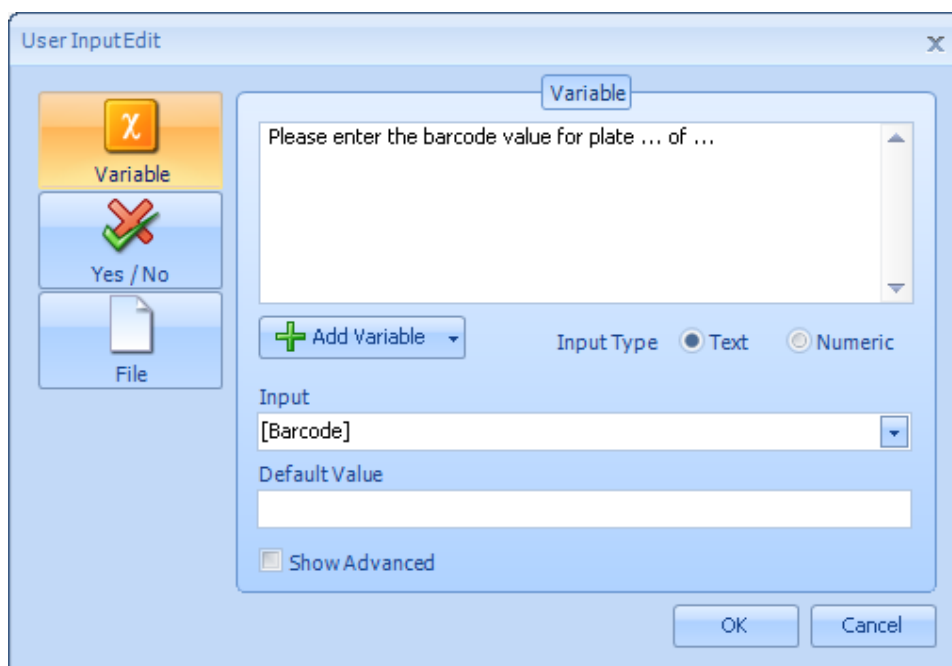
The **User Input** message box allows the person setting up the system to ask the user to enter information for later use in the run. An example would be to elicit how many plates in a run.

The edit box allows the information to be entered:



### The User Input screen

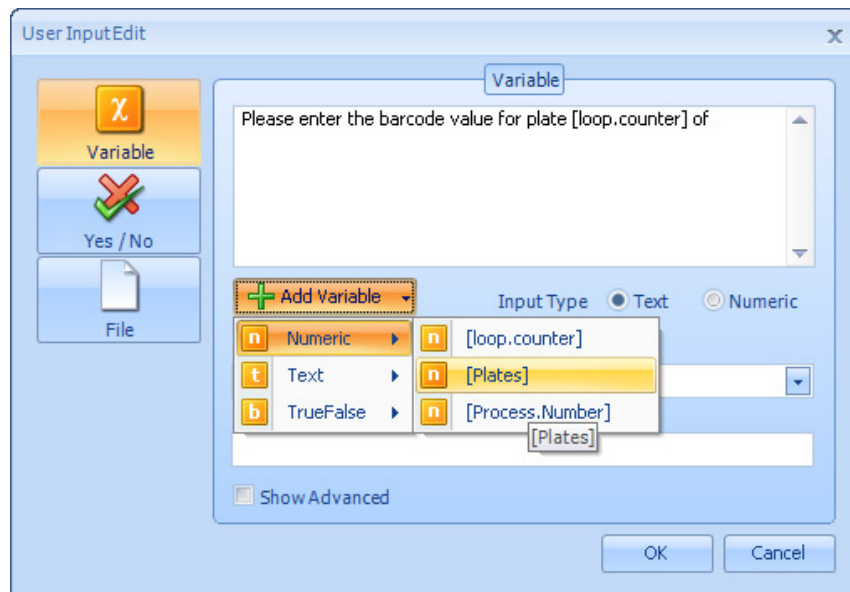
The user can be asked to enter a value, a yes/no answer or for a filename. For the entry of a variable, enter the question that the user will be asked in the message text box.



### Enter the message to be displayed at runtime

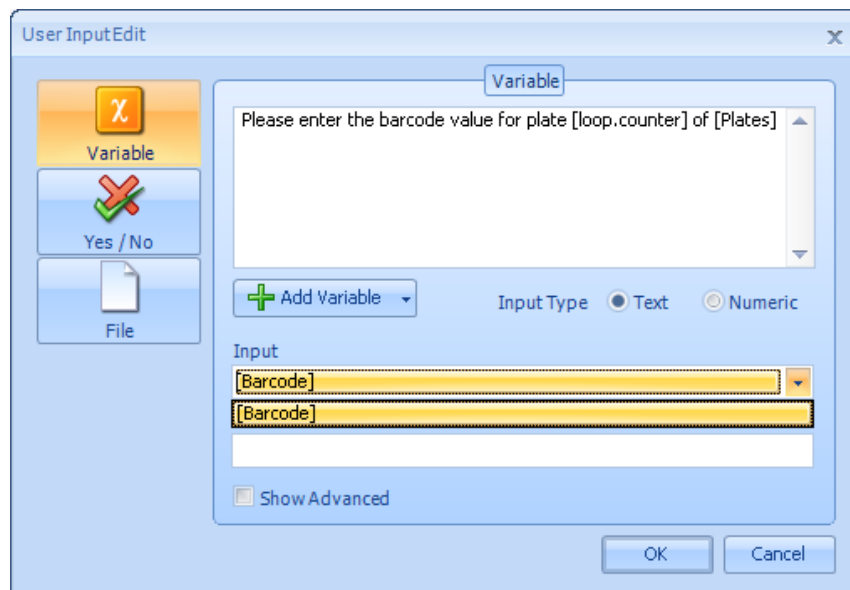
As with a message box, a variable can be requested as part of the message if required using the **Add Variable** option. The user can also be prompted for a textual value or numeric value with the radio button. In this example we are going to ask the user for the barcode value for the second plate of six plates in the stack. The plate value (2) will be stored in **[loop counter]** the total number of plates in

**[plates].** Once the message has been entered, use the add variable button to insert the two variables in the message:



## Add a variable to the message

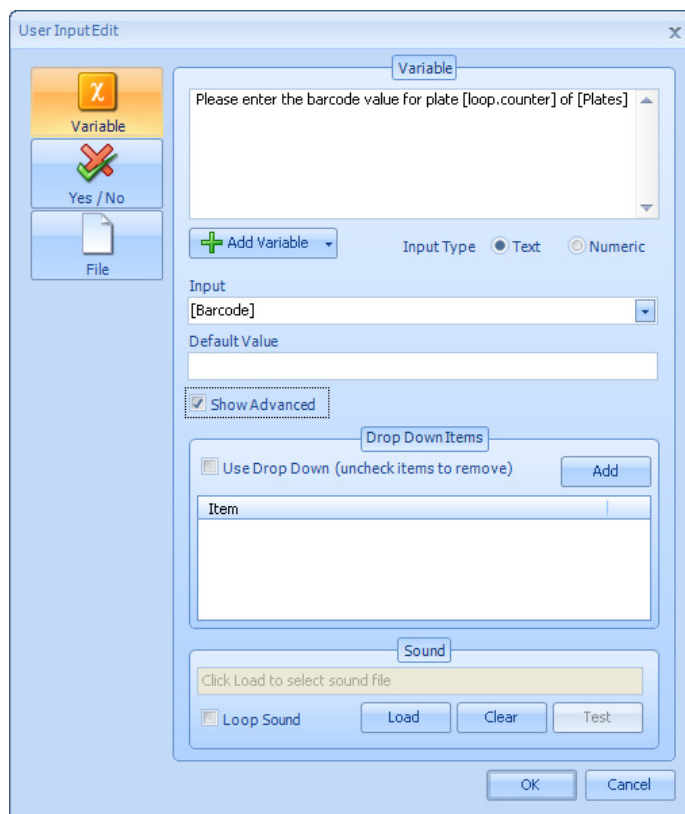
A barcode is of type **Text**, the select which variable will be loaded with the input value at runtime.



## Select the variable to receive the input at runtime

A default value can be entered or the user to see at runtime if there is a usual value for the input. If the OK is selected, and the procedure saved and run, the input message will be displayed.

As for Message Box, sounds can be entered to be played at runtime. Click the **Show Advanced** check box and select the AVI file to be played at runtime.



**Adding a sound recording to the User Input screen**

## 4.10 Scripts

Scripts allow code snippets of VBScript, VB.NET or C# to be run as part of a Procedure. Using scripts allows custom code to be used to perform actions which are very specific to the application being automated. These actions include

- Formatting Plate Data so that it can be uploaded to a LIM
- Running Stored Procedures on an SQL Database and using this data to populate Overlord variables
- Generating Pipetting Worklists based upon the barcode of a plate.
- Sending status messages via a 3<sup>rd</sup> Party application (e.g. SMS to a mobile phone).

These are just some of the actions Scripts can carry out. The list is endless. Whatever a piece of code can do inside Visual Studio or as part of an Excel macro, it pretty much can do via a script. The only exception is that it cannot display windows, message and input boxes being the only exceptions. However, there are ways around this which will be described in this section

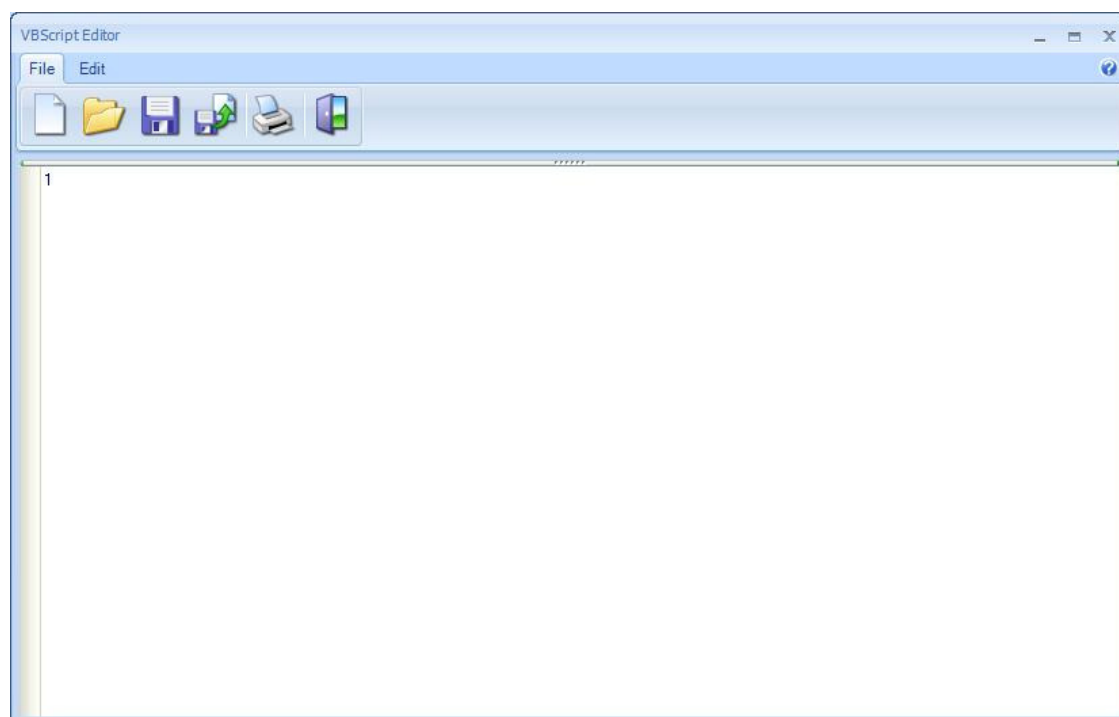
For more information on the code syntax on each language that is supported in Scripts, please take a look at the following resources.

Language	Title	Address
C#	C# Programming Guide	<a href="http://msdn2.microsoft.com/en-us/library/67ef8sbd(VS.80).aspx">http://msdn2.microsoft.com/en-us/library/67ef8sbd(VS.80).aspx</a>
C#	C# Reference	<a href="http://msdn2.microsoft.com/en-us/library/618ayhy6(VS.80).aspx">http://msdn2.microsoft.com/en-us/library/618ayhy6(VS.80).aspx</a>
Visual Basic	Visual Basic Programming Guide	<a href="http://msdn2.microsoft.com/en-us/library/y4wf33f0(VS.80).aspx">http://msdn2.microsoft.com/en-us/library/y4wf33f0(VS.80).aspx</a>
Visual Basic	Visual Basic Reference	<a href="http://msdn2.microsoft.com/en-us/library/25kad608(VS.80).aspx">http://msdn2.microsoft.com/en-us/library/25kad608(VS.80).aspx</a>
VBScript	VBScript Users Guide	<a href="http://msdn2.microsoft.com/en-gb/library/sx7b3k7y.aspx">http://msdn2.microsoft.com/en-gb/library/sx7b3k7y.aspx</a>
VBScript	VBScript Reference Language	<a href="http://msdn2.microsoft.com/en-gb/library/d1wf56tt.aspx">http://msdn2.microsoft.com/en-gb/library/d1wf56tt.aspx</a>

There is also a set of VBScripts that are commonly used in Overlord, available from the PAA website. Assuming you have logged in, the scripts are available from <http://www.paa.co.uk/labauto/knowledgebase/vbscripts.asp>

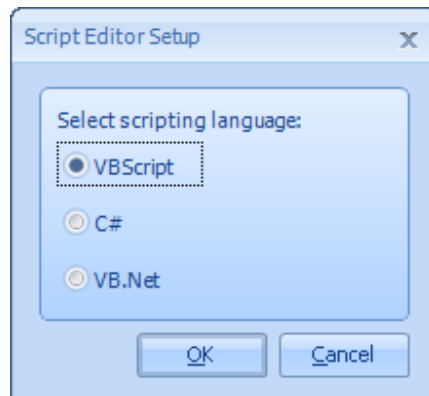
The following section assumes you are familiar with VBScripts. If not, please look at the resources listed above.

When the script command is added to a procedure, the following window will appear.



**New Script**

The Script Editor, as this window is called, will start up using the language that has been set as default in the Setup for the Script. Setup for Scripting can be reached by clicking on the small arrow to the right of the Script icon in the command palette.



**Script Setup**

The language can also be changed by clicking the New toolbar button in the Script Editor. Before loading up a new code pane, a dialog similar to that in Setup will appear asking what language should be used.

Once the language has been set, code can be entered into the pane. The following lines of code would rename a data file.

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
objFSO.CopyFile "C:\Data.txt", "C:\Result.txt", true
```

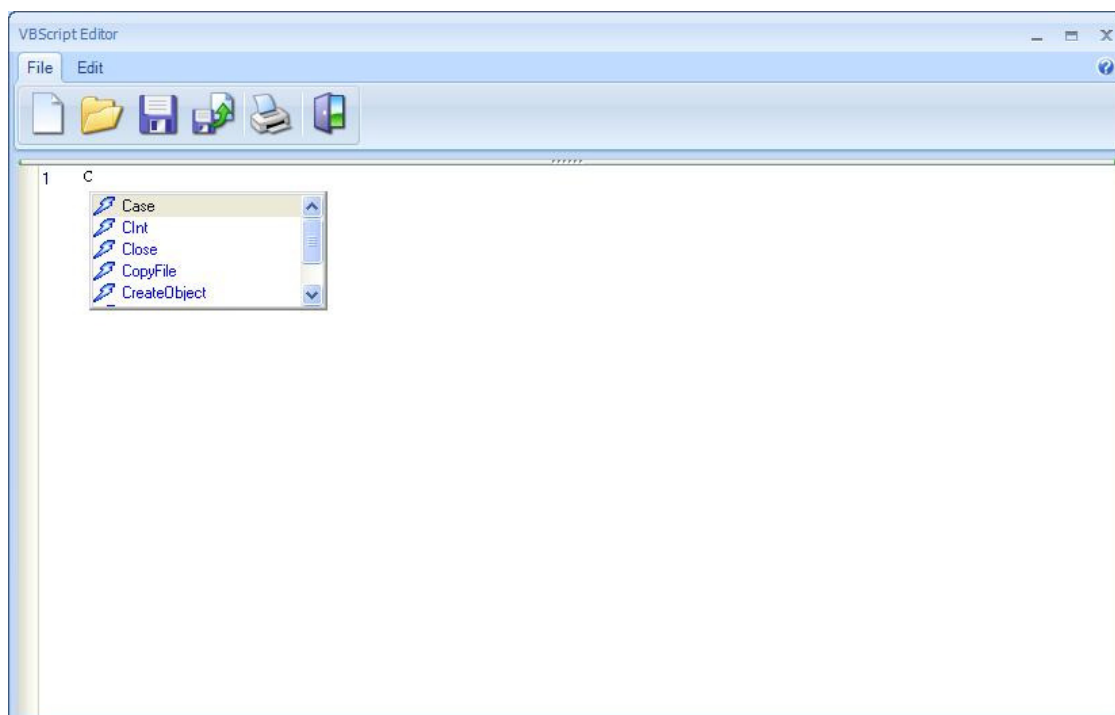
Save this code to a file, in this case a VBS file and then exit. The code will now be part of the procedure, although it is important to note the procedure refers to the file that contains to the code, not the code itself.

At runtime, this code would execute a copy the file Data.txt to Result.txt.

## 4.10.1 IntelliSense

The Script Editor supports IntelliSense, which is the ability to autocomplete a keyword that is being entered. If for example the letter 'C' is typed into the Code pane and then CTRL-SPACE (the Control and Space button on the keyboard) is then pressed, the IntelliSense menu will come up with all the keywords that begin with C.





### IntelliSense

Selecting the entry in this menu will then enter this keyword into the Code pane. The IntelliSense menu will only appear if the text entered cannot be resolved into a name. If it can, the keyword will automatically be filled in. For example, if **CreateO** was entered followed by pressing the key CTRL-SPACE on the keyboard, the keyword **CreateObject** would be completed automatically. It is always clear that Intellisense recognizes a keyword as when it is completed, it will be highlighted in blue. It should also be noted that pressing CTRL-SPACE where the cursor is not beside any text will bring up the Intellisense menu with every keyword for the selected language.

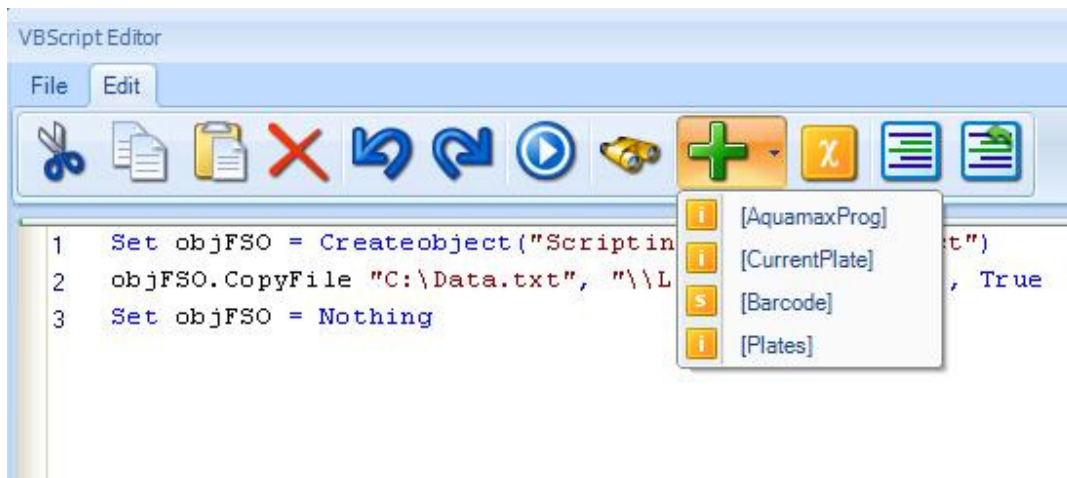
## 4.10.2 Using Overlord variables in Scripts

A more useful script would be one that takes the data just generated by a plate reader (Data.txt) and renames it with the barcode of the plate and stores this on a network drive (e.g. \\LIMS\AssayData\). The following code would do this

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
objFSO.CopyFile "C:\Data.txt", "\\LIMS\AssayData\" + [Barcode] +
".txt", true
```

The difference here to the previous example is that the variable [Barcode] has been used for the new name of the file. It is assumed that the variable [Barcode] would contain the barcode of the plate on the reader when this script is run. If [Barcode] contained the value **"B12345"**, then the file would be copied to \\LIMS\AssayData and would be called **"B12345.txt"**

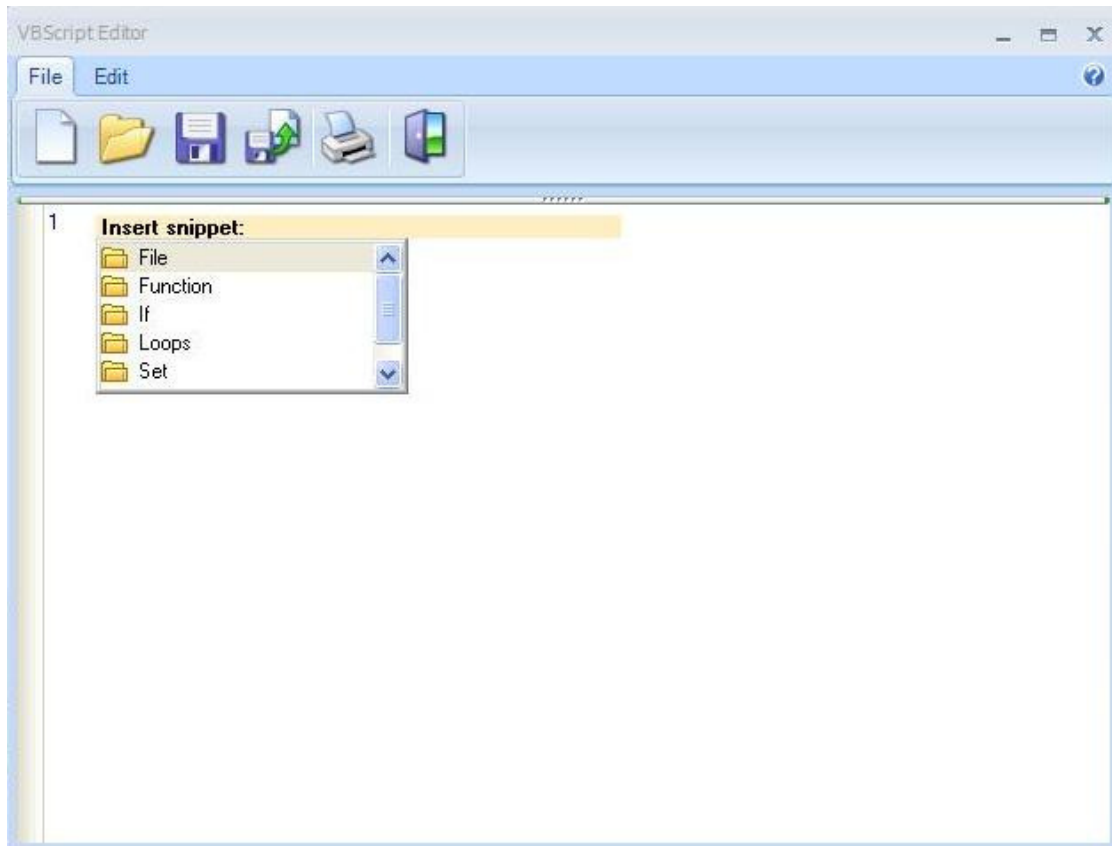
Variables can be added to the location of the cursor in the code pane by clicking on the Edit section of the Toolbar in the Script Editor and then clicking on the 'Insert Variable' button. If a new Overlord variable is required at this point, the 'Add Variable' button next to 'Insert Variable' can be clicked which will allow a new variable to be added.



### Add Variable

#### 4.10.3 Code Snippets

Code snippets are blocks of common functions that can be added into any part of a script and be easily customized for the task in hand. Right mouse clicking on the menu brings up the context menu and at the bottom is 'Insert Code Snippet'. Selecting this will bring up a sub menu of available snippets



### Insert Snippet

Each folder in this list contains different categories of snippet type. Selecting the 'Set' folder would allow the 'Set Statement' entry to be selected. Once this is selected, by pressing ENTER on the keyboard, the snippet will be inserted into the Code pane.

```
1 Set objFSO = Createobject("Scripting.FileSystemObject")
```

When a snippet is added, any additional text that is required to be entered to customize the snippet will be surrounded by a green box or highlighted in light red. In the example above, the text **objFSO** and **Scripting.FileSystemObject** could be modified. The part of the snippet that can first be changed is highlighted in light red. The current text **objFSO** is fine so we can move on to the next part by pressing ENTER on the keyboard. The highlight will then move onto **Scripting.FileSystemObject** so that this can be changed if necessary. Again, in this case, the current text is fine and pressing ENTER again will confirm the snippet is complete.

Now an instance of the **FileSystemObject** has been created (or instantiated), it can be used. In Overlord scripts this is commonly used to work with files as seen in the examples above. Selecting the code snippet File → CopyFile Method will add this method call which can be against the **objFSO** object.

```
1 Set objFSO = Createobject("Scripting.FileSystemObject")
2 objFSO.CopyFile Source Path, Destination Path, True
```

T

this time the default text entries on the snippet will need modifying as follows

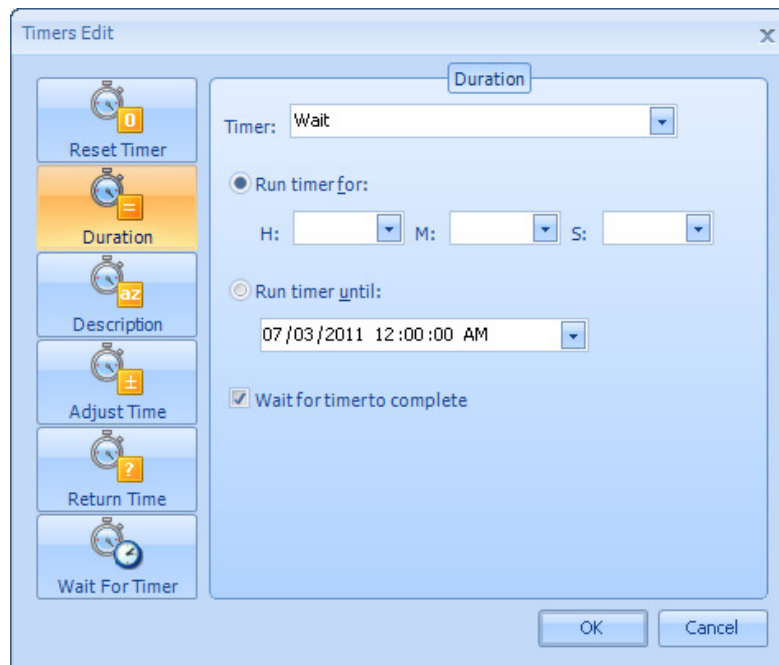
- Name → objFSO
- Source Path → "C:\Data.txt"
- Destination Path → "\\LIMS\PlateData\" + [Barcode] + ".txt"
- True (Overwrite?) → True

With the snippet complete another line can be typed in this time that sets the object **objFSO** to equal nothing. We now have the same code that was used in the example in Section 4.10.2 Using Overlord variables in Scripts. Using code snippets allows scripts to be written much quicker and also reduces the chances of syntax errors since code, that is known to work, is used.

```
1 Set objFSO = Createobject("Scripting.FileSystemObject")
2 objFSO.CopyFile "C:\Data.txt", "\\LIMS\PlateData\" + [Barcode] + ".txt", True
3 Set objFSO = Nothing
```

### 4.11 Timers

The Timers procedure step allows the user to manipulate time during the run. The edit window is shown below:



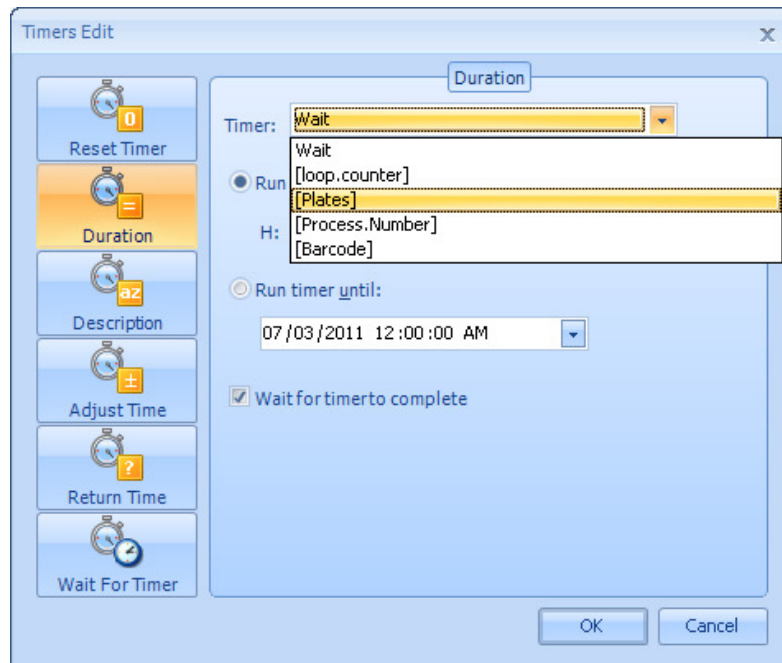
**The Timers edit panel**

There are 5 options:

- Duration; set the timer
- Description;
- Adjust Time;
- Return Time;
- Wait For Timer

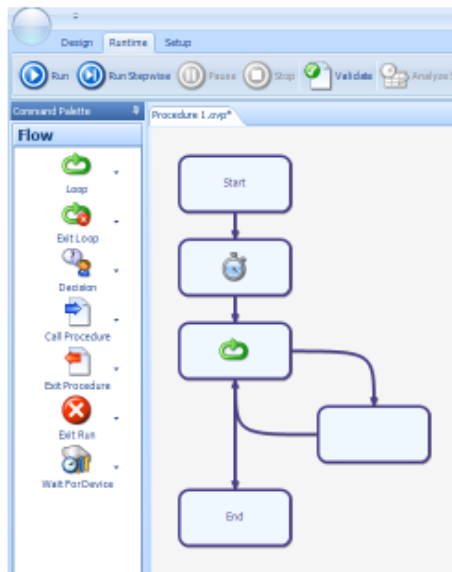
#### 4.11.1 Duration

Select the variable to be set as the timer:



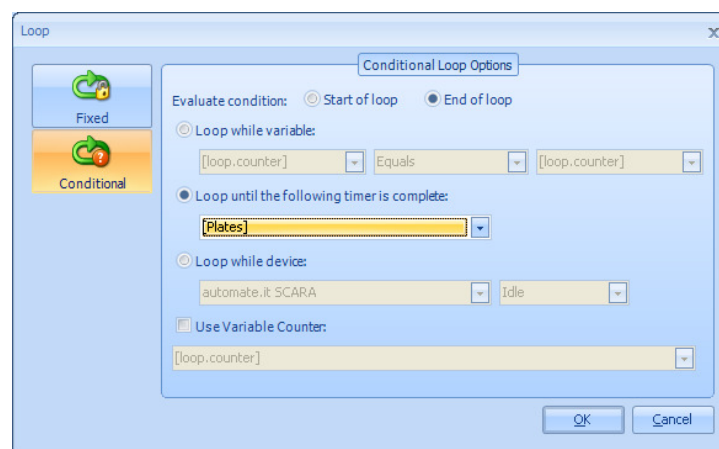
#### Choosing the variable to use as the timer

The user should then select either **Run timer** for or **Run time until** to set the timer duration. The timer can be an absolute time or a timer determined by the variables in the drop down list. To detect the end of the timer period either check the **Wait for timer to complete** option or use a loop with the timer option set with the timer variable selected:



**Set a timer and then wait for it to elapse in the loop**

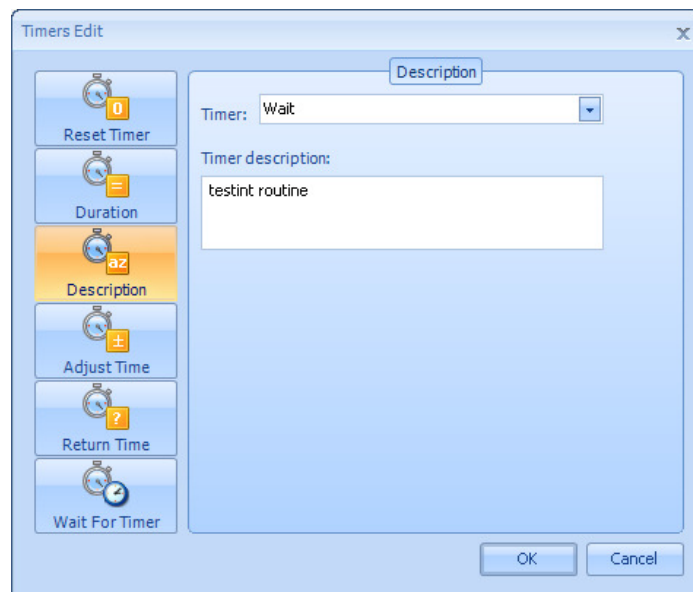
And the individual loop procedure step looks like this:



**Loop test for the end of the run**

#### 4.11.2 Description

This allows the user to define a name for the timer. This is advisable when there are multiple timers in a run.

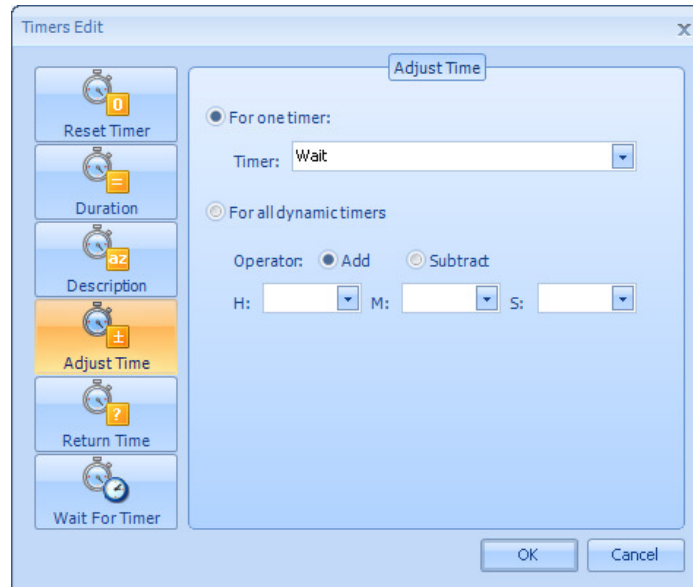


**Add a descriptive name to the timer**



### 4.11.3 Adjust Timer

This allows the user to make changes to a timer setting during a run.

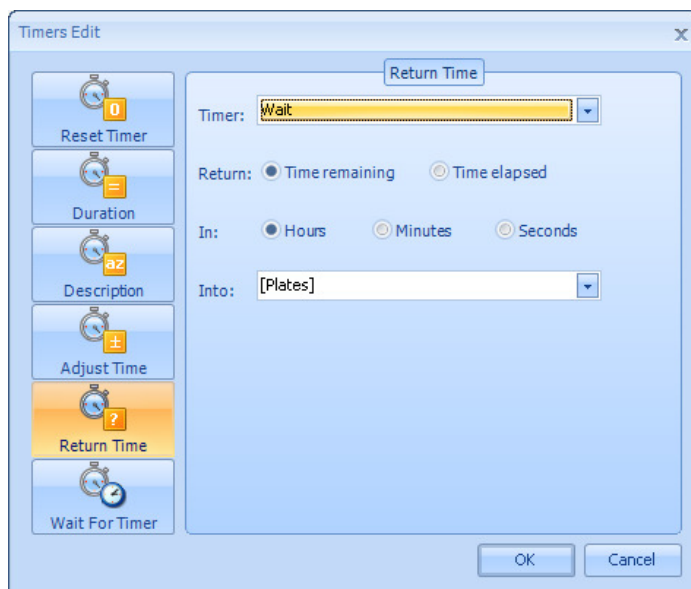


**Adjust time option**

The user should choose whether one timer (**For one timer**) or all the timers (**For all dynamic timers**) will be affected by this procedure step. Then the user should then decide if the timers should be increased in time (**Add**) or decreased (**Subtract**).

#### 4.11.4 Return Time

This allows the user to get a time value into a variable for manipulation.

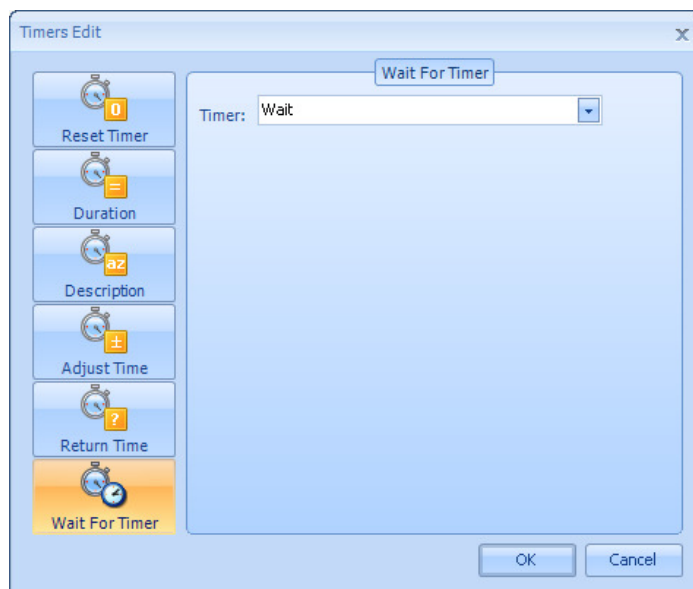


**Return time edit screen**

Select the timer that the value is required, choose the **Time remaining** or **Time elapsed** option, the unit for the time to be reported and finally the variable to which the information is to be assigned.

### 4.11.5 Wait for Timer

This allows the user to halt the **Overlord2** procedure until a selected timer has elapsed.



**Wait for a timer to elapse**

Select the timer to wait to elapse.

## 4.12 Loop

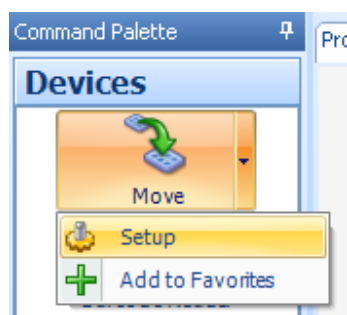
This allows the user to halt the **Overlord2** procedure until a selected timer has elapsed.

## 4.13 Move Command

The Move Command provides a simple interface to allow basic users to move Labware around a system, without having to define the steps required in detail. The system should be setup by an advanced user, but once this is complete changes are only required if the system changes, for example if a new instrument is added.

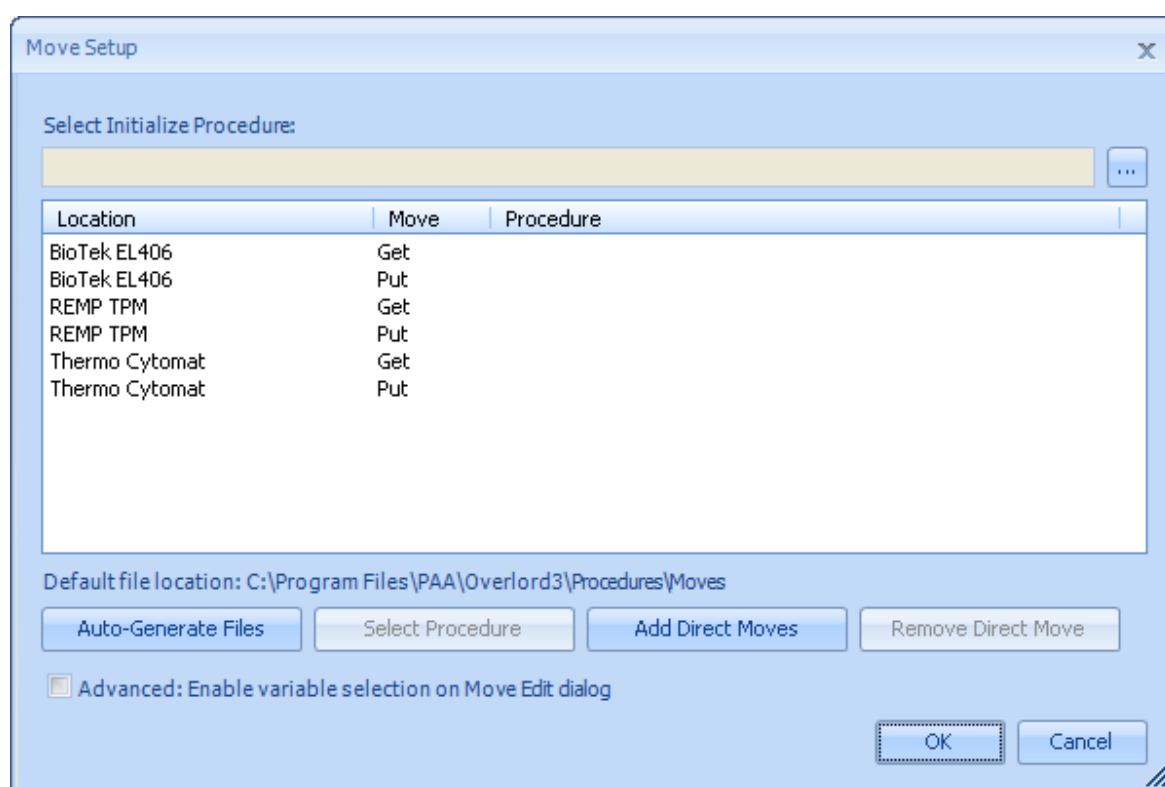
### 4.13.1 Setup

The Move Command is used in a Standard or Schedule Procedure the Setup dialog must be completed. To open the Setup dialog select **Devices** from the **Command Palette**, click the right-hand part of the **Move** button, then click **Setup** (Select Move Command Setup):



**Select Move Command Setup**

The **Move Setup** dialog will appear. Initially two entries will be listed for each device on the system (Move Setup Initial State). Each device will have a **Get** and a **Put** type entry.



**Move Setup Initial State**

The Move Command works by linking pre-defined Procedures (Move Command Procedure - .mcp files) together at runtime to move a Labware item from one location to another. The following explanation uses the example of moving Labware from a Thermo Cytomat storage device to a BioTek EL406.

## 4.13.2 Move Type

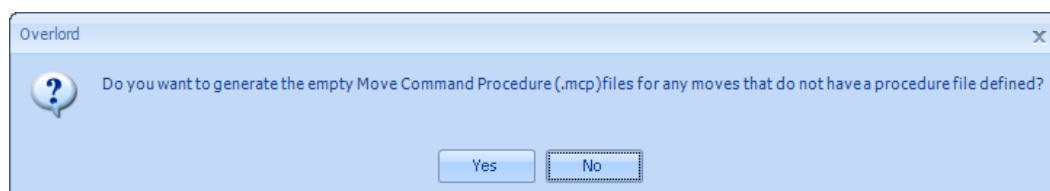
The entries in the Move Setup list can be one of three types: Get, Put or Direct Move. By default each device on the system has a Get and Put entry. Direct Move entries are for advanced use and not required on all system. They are explained in more detail later in the section.

**Get** – A Get Move Command Procedure should contain the Actions necessary to move Labware from the initial storage location (e.g. Thermo Cytomat) to a safe point. The Procedure should end with the Labware being gripped by the robot, ready to move to the destination location.

**Put** – A Put Move Command Procedure should contain the Actions necessary to move Labware from a safe point, with the Labware already gripped by the robot, to the destination location (e.g. BioTek EL406).

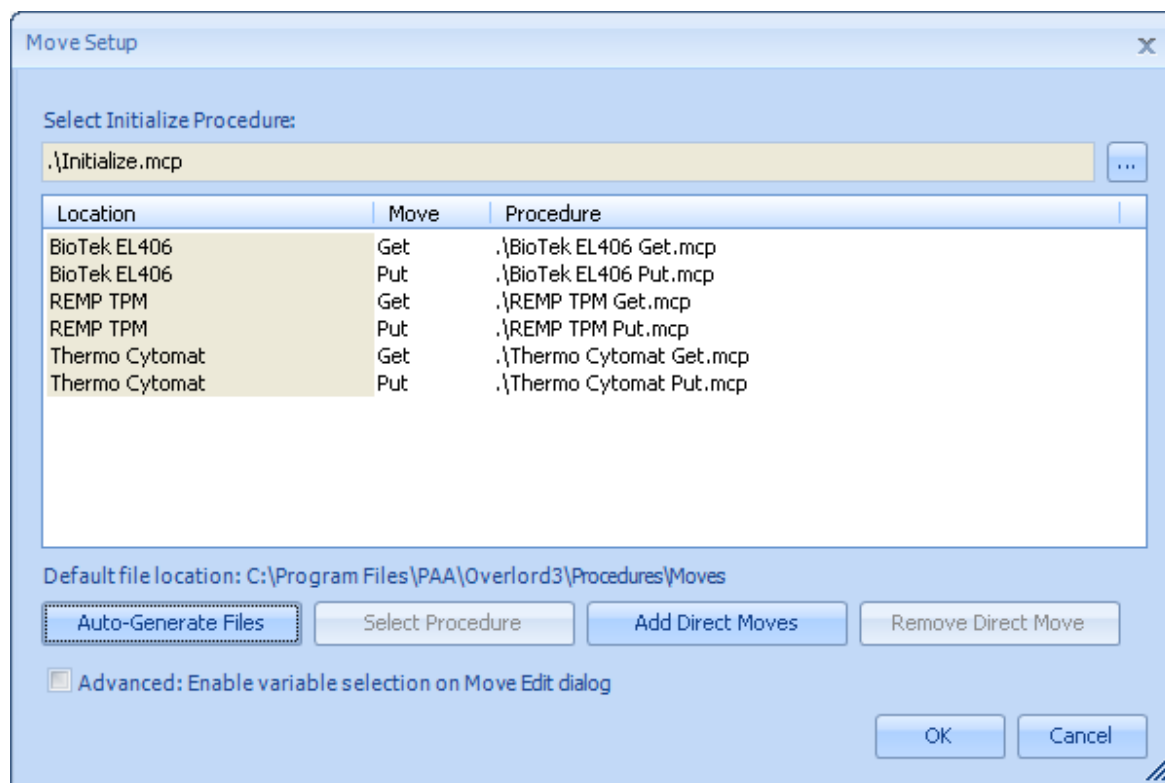
**Direct Move** – A Direct Move, Move Command Procedure, should contain the Actions necessary to move Labware from the initial storage location (e.g. Thermo Cytomat) to the destination location (e.g. BioTek EL406).

The empty Procedures for all Moves can be generated automatically. Click the **Auto-Generate Files** button. A dialog will appear to check that the user wants to go ahead and generate the files (Auto-Generate Files). Click **Yes** to continue and generate files.



**Auto-Generate Files**

Once the files have been generated the Setup dialog will be updated to show the names of the auto-generated Procedures (Auto-Generated Files Displayed).



Auto-Generated Files Displayed

### 4.13.3 Initialize Procedure

This defines the Procedure that is called at runtime if the **Move – Initialize** action is called. It provides a way to make sure users initialise all devices on a system with only one Action.

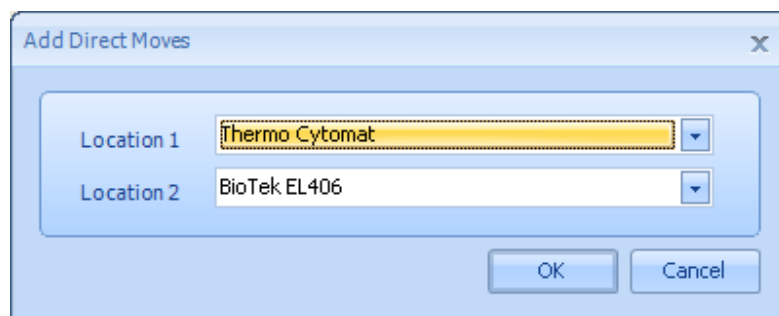
### 4.13.4 Change Selected Procedure

To change to Procedure that is used for a particular Move, either double-click the entry, or select the entry and click the Select Procedure button. A file browse dialog will then allow any .mcp file to be selected.

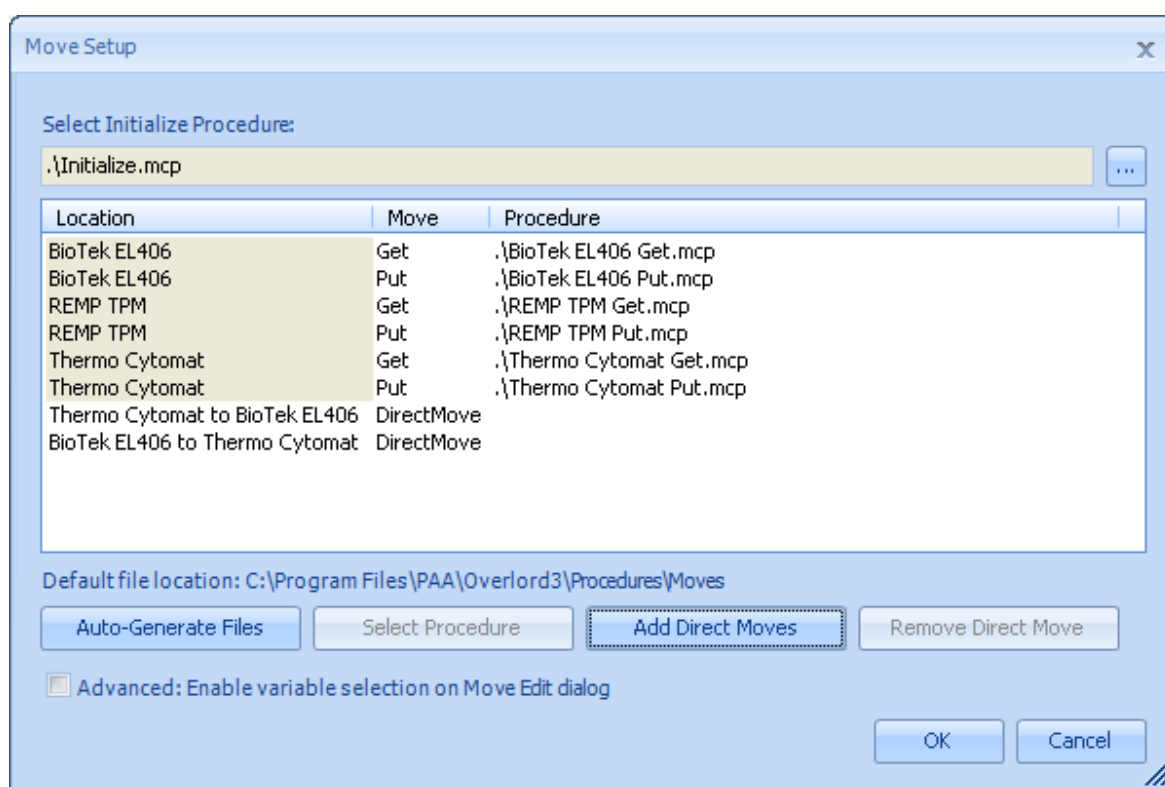
### 4.13.5 Direct Moves

The default behaviour for a system is to link Get and Put moves together. This is done so that we do not have to define every combination of moves on the system. However, in certain cases it may be desirable to define a single Procedure to move Labware to one location to another. An example of this is a Stacker device, where the Stacker Action moves the Labware in one Action.

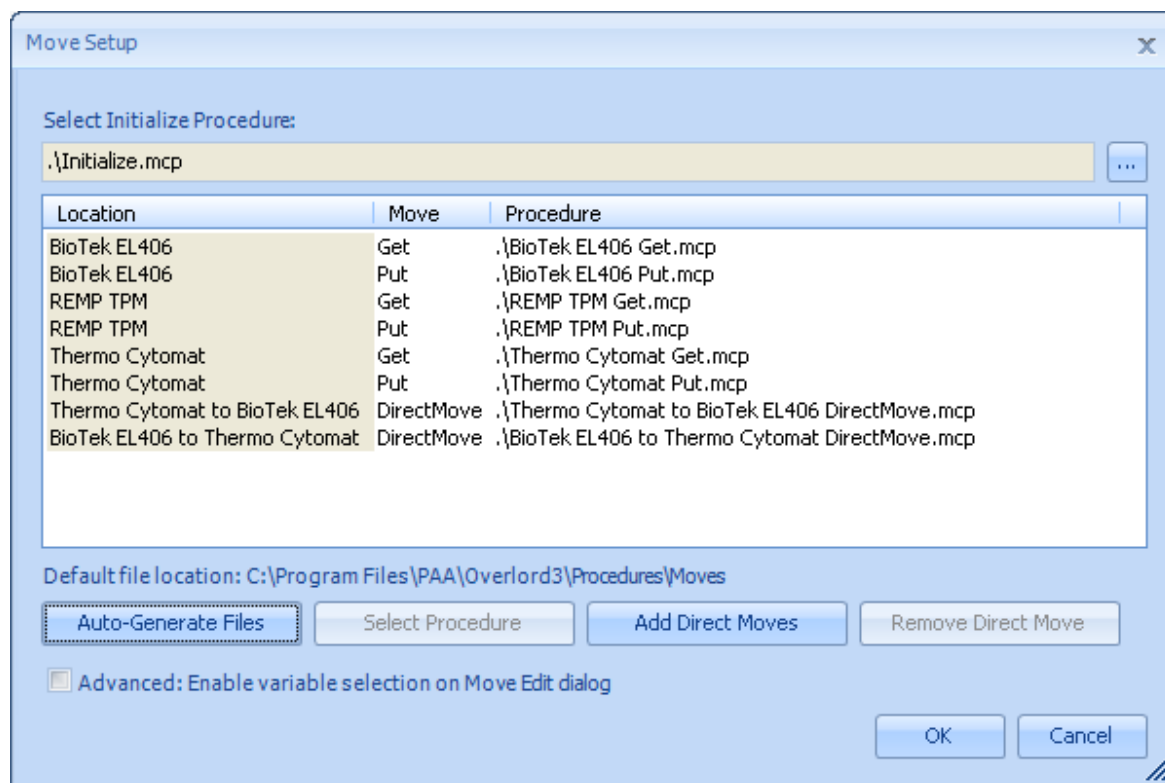
To define a Direct Move, click the **Add Direct Moves** button. The **Add Direct Moves** dialog will appear (Add Direct Moves):

**Add Direct Moves**

Select the two locations required for the move. Click **OK**. Two entries will be generated in the move items list, one for each direction of travel – A to B and B to A (Direct Moves Added):

**Direct Moves Added**

Direct Move Procedures can be generated in the same way a Get and Put Procedures, using the **Auto-Generate Files** button (Auto-Generate Direct Move Files).



#### Auto-Generate Direct Move Files

When a Direct Move has been defined this will override the Get and Put Procedures for that specific move at runtime.

In the example, if the user selects a move from Thermo Cytomat to BioTek EL406, at runtime a single Procedure will be run – **Thermo Cytomat to BioTek EL406 DirectMove.mcp**. However, when the user moves Labware from the Thermo Cytomat to the REMP TPM it will link two Procedures – **Thermo Cytomat Get.mcp**, followed by **REMP TPM Put.mcp**.

A Direct Move entry can be removed by selecting the entry in the list and clicking **Remove Direct Move**.

#### 4.13.6 Advanced: Enable variable selection on Move Edit dialog

This is an advanced feature that adds Overlord Text Variables to the Labware, From Location and To Location drop-down lists on the Move Command Edit dialog. Tick the box to enable to feature. The feature is explained in further detail in the Edit dialog section.

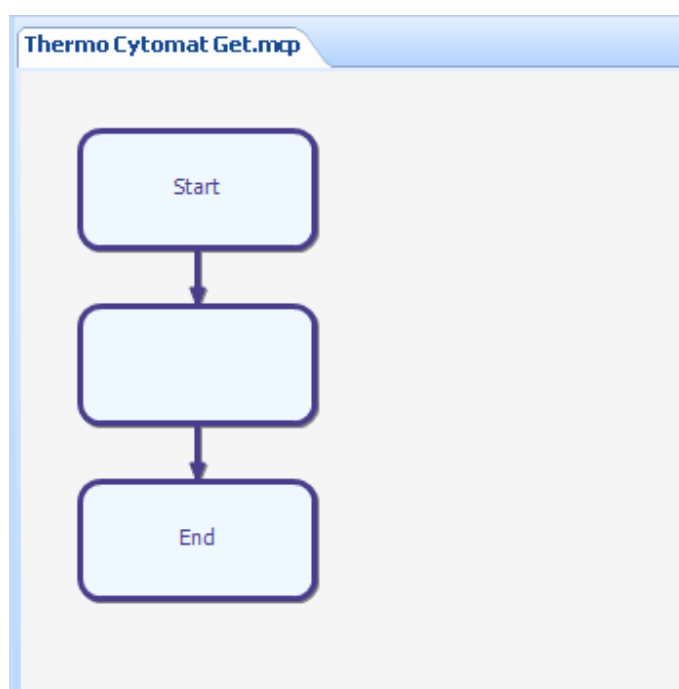


## 4.14 Move Command Procedures

The Move Setup section explained how to define which Move Command Procedures to use for each Device Get and Put. The next step of the configuration process is to complete the Procedures created using **Auto-Generate Files**.

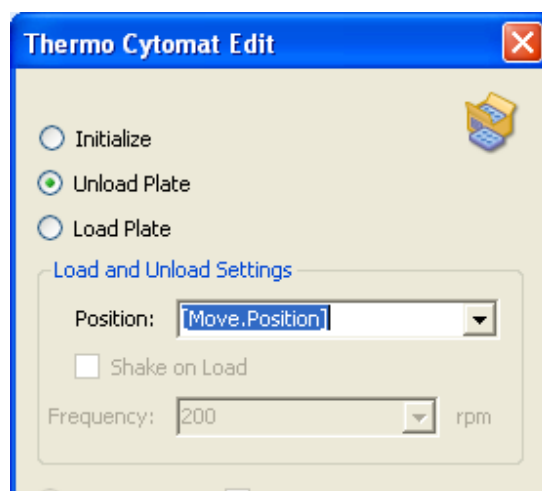
### 4.14.1 Get Move Command Procedure

Open a Move Command Procedure Get file. By default Move Command Procedures are generated in a sub-directory of the Procedures directory called Moves. The Procedure should be empty (Empty 'Get' Move Command Procedure):

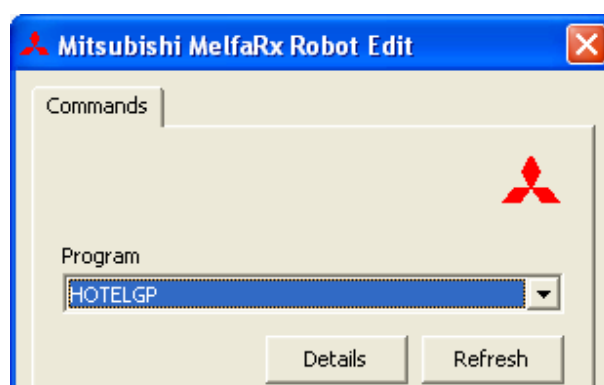


**Empty 'Get' Move Command Procedure**

Add Actions to the Procedure to get the Labware from the storage device, pickup with robot and move to safe point. In the example, this means adding an unload Cytomat Action (Cytomat Unload Action), followed by a robot Action to pick up the Labware Robot (Get Labware From Hotel Action). It is important to use the variable **[Move.Position]** in any Storage Device position related Actions. This is because the value of [Move.Position] is set at runtime based on the system configuration.

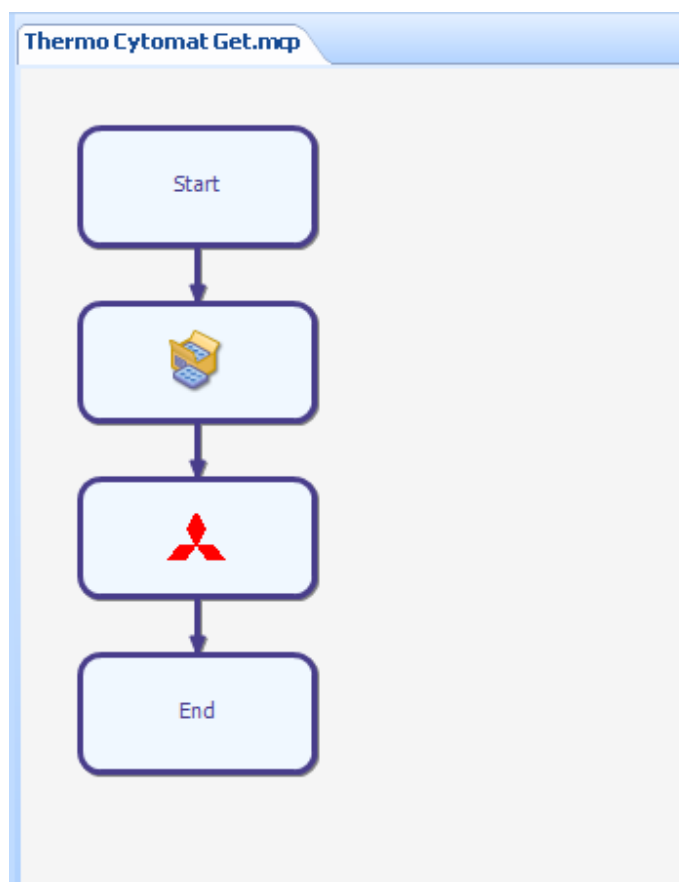


**Cytomat Unload Action**



**Robot Get Labware From Hotel Action**

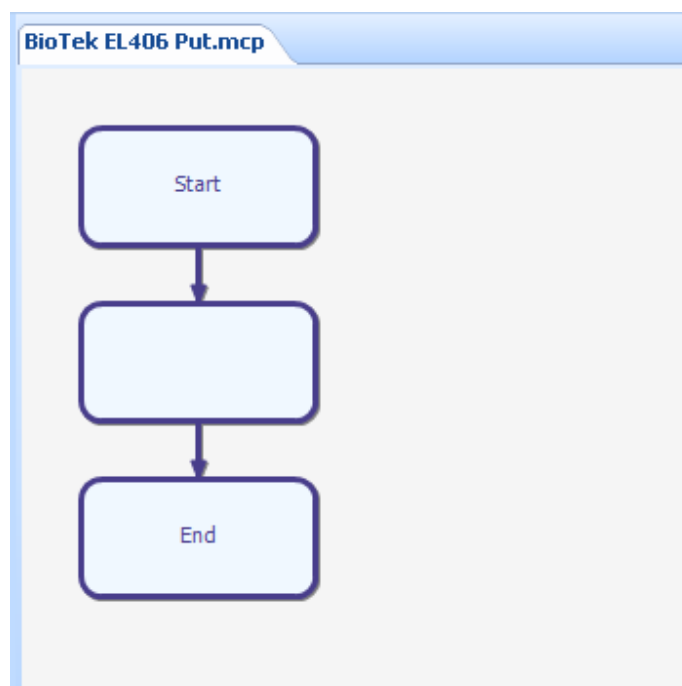
The Procedure should now resemble **Completed 'Get' Move Command Procedure**. Make sure the Procedure is saved:



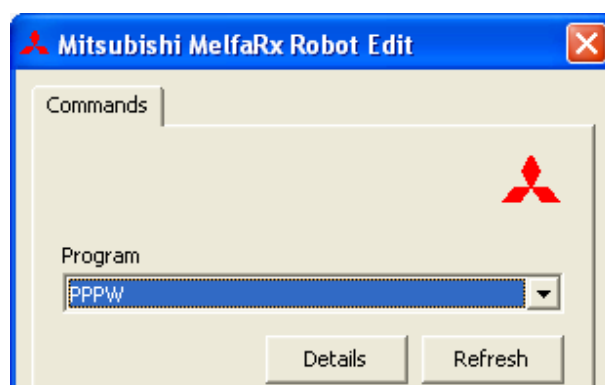
**Completed 'Get' Move Command Procedure**

#### **4.14.2 Put Move Command Procedure**

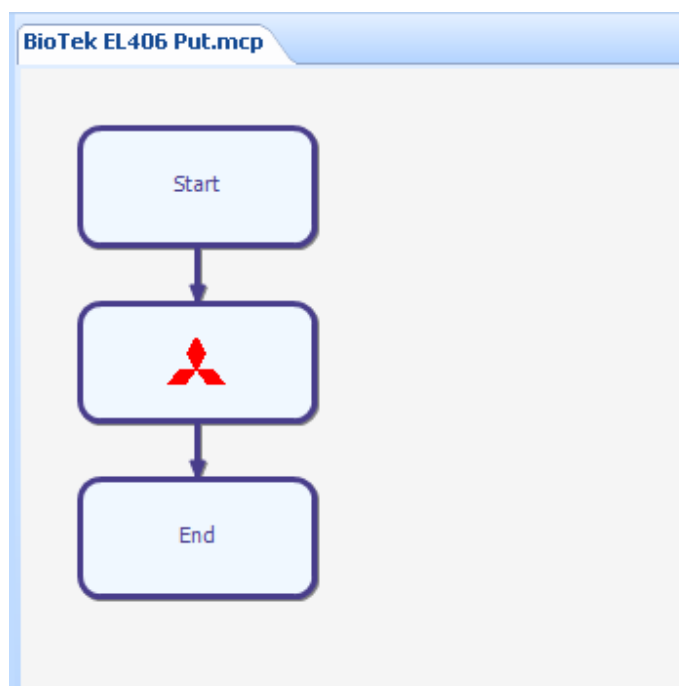
Open a Move Command Procedure Put file. In this example we will complete the Procedure to move the Labware to the BioTek EL406. This is simpler than the Storage Get procedure as only the robot Action to put the Labware on the device is required.



**Empty 'Get' Move Command Procedure**



**Robot Put Labware On Device**

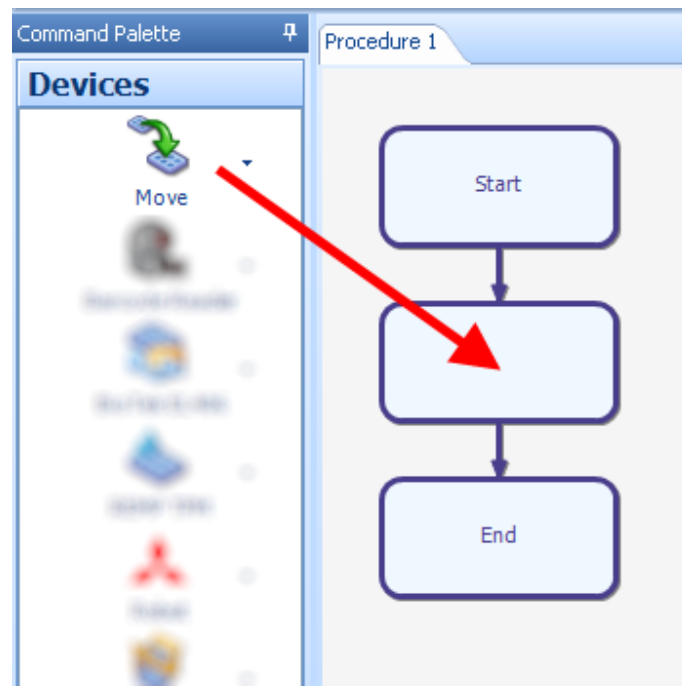


**Completed 'Put' Move Command Procedure**

The Move Command Procedures should be completed for all Get and Put moves. Any Direct Moves Procedures can be completed in the same way, making sure the Labware is moved completely from the source to the destination location.

#### ***4.15 Adding a Move to a Standard or Schedule Procedure***

Once the Move Setup and all required Move Command Procedures have been defined Move Actions can be added to Standard or Schedule Procedures. To add a Move Action to a Procedure drag-and-drop the Move onto the Procedure flow from the **Devices** pane of the **Command Palette** (Drag-and-drop Move Action).

**Drag-and-drop Move Action**

The Move Edit dialog will appear (Move Edit). Select the **Labware** move, along with the **From** and **To** locations.

The Labware listed is defined in Labware Setup (see Labware Setup section).

The From and To locations list either the device alias or the Positions Set alias, both set via Command Setup. Where a device has more than one position defined the Position item allow selection of the position. However, when a device is a robot, stacker or storage device the Position item is disabled, as the position will be set atomically at runtime.

Move Edit

☐ Initialize  
☒ Move

Process number [disabled]

Get the labware Plate96

From the Thermo Cytomat

Position 0

and move to BioTek EL406

Position 1

☐ Return barcode to [Barcode]

Advanced (Storage devices during Process running only)

☐ Use wait time (seconds)  
600

OK Cancel

Move Edit

#### 4.15.1 Process Number

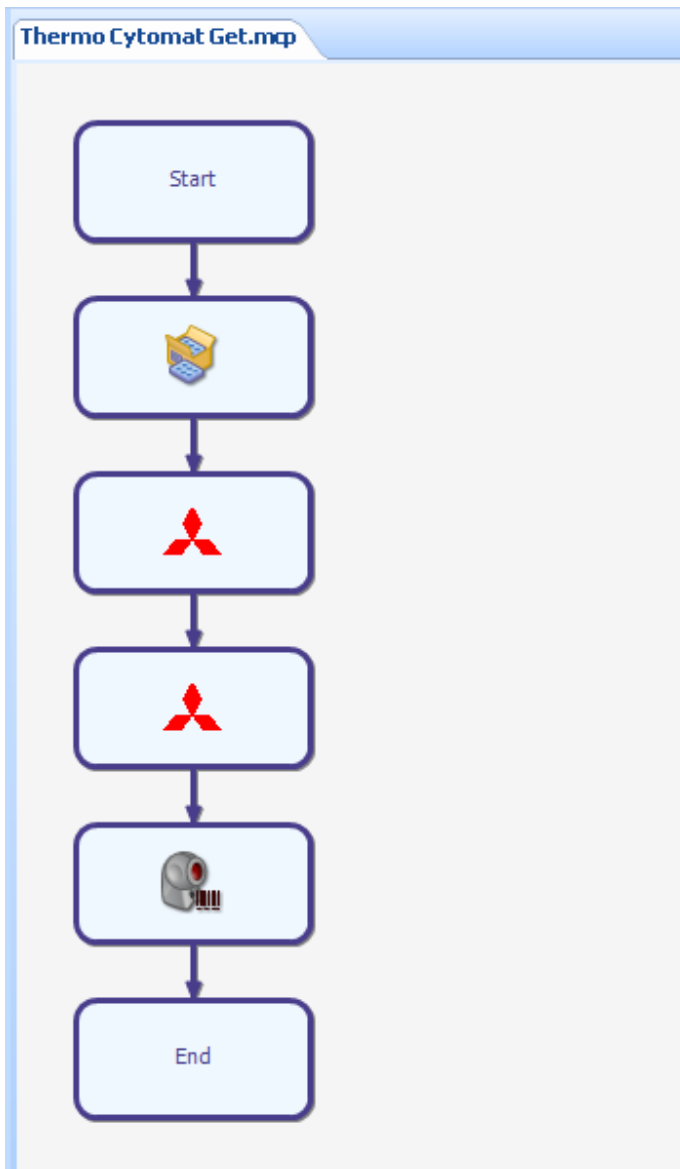
When a Move Action is added to a **Schedule Process** the **Process number** option is **disabled** as this value is calculated automatically at runtime.

When a Move Action is added to a **Standard Procedure** the **Process number** option is **enabled** and an Overlord Numeric Variable must be selected. At runtime this variable must be set to the Process you want to carry out the move for. This is important for moves from and to storage devices, so that the position to unload ([Move.Position]) can be calculated correctly.

#### 4.15.2 Return Barcode

The **Return barcode to** option is only enabled if the Get Procedure contains an Action that uses the **[Move.Barcode]** variable. This means that the option can only be selected when the Procedure will return a barcode.

For example, from the earlier example of Thermo Cytomat Get.mcp, adding an additional robot action to move the Labware past a barcode reader, followed by a Barcode Reader action to return the barcode to [Move.Barcode] ('Get' Move Command Procedure To Return Barcode), will enable to option on the Move Edit dialog ( 'Return barcode to' Option Enabled)



**'Get' Move Command Procedure To Return Barcode**



Move Edit

☐ Initialize  
☒ Move

Process number [ ]

Get the labware Plate96 [ ]

From the Thermo Cytomat [ ]

Position 0 [ ]

and move to BioTek EL406 [ ]

Position 1 [ ]

☒ Return barcode to [Barcode] [ ]

Advanced (Storage devices during Process running only)

☒ Use wait time (seconds)

600 [ ]

OK Cancel

**'Return barcode to' Option Enabled**

Tick the box and select a variable to return the barcode value.

### 4.15.3 Use Wait Time

The **Use wait time** option is available only for Storage devices (including robot and stackers) during Scheduled Processes. To enable the option the move To location must be a storage device (Use Wait Time). The Use wait time box can then be ticked and a duration, in seconds, entered. At runtime, once the plate has been loaded into the storage device the system will wait for the given duration before the next Move Action is executed for the process.

Move Edit X

☐ Initialize

☒ Move

Process number

Get the labware

From the

Position

and move to

Position

☐ Return barcodes to

Advanced (Storage devices during Process running only)

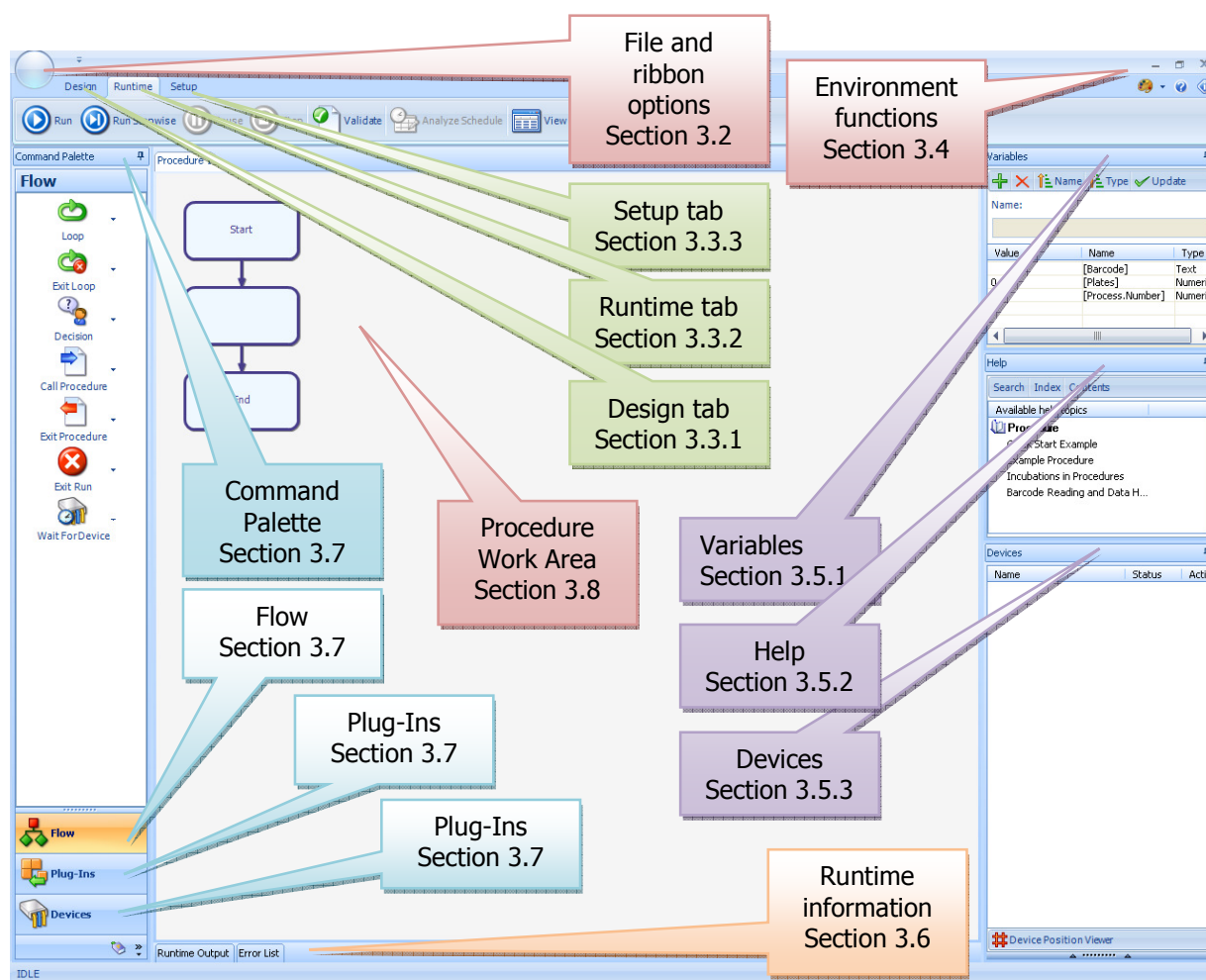
☒ Use wait time (seconds)

## Use Wait Time

## 5 Main Overlord3™ screen

### 5.1 Running Overlord3™

When Overlord3 is launched from the start menu, file explorer or desktop, the Overlord3 main screen is displayed (Main Overlord3 screen):



**Main Overlord3 screen**

The Overlord3 main screen has the following areas (clockwise from top left):

- File options (Section 5.2 File ribbon options)
- Tab area (Section 5.2.2)
- Design (Section 5.3.1)
- Runtime (Section 5.3.2)
- Setup (Section 5.3.3)
- Environment functions (Section **Error! Reference source not found.**)
- Variables (Section 5.4.1)
- Help (Section 5.4.2)

- Devices (Section 5.4.3)
- Runtime information (Section 0)
- Command palette (Section 5.6)
- Flow
- Plug-Ins
- Devices
- Procedure work area (Centre, Section 0)

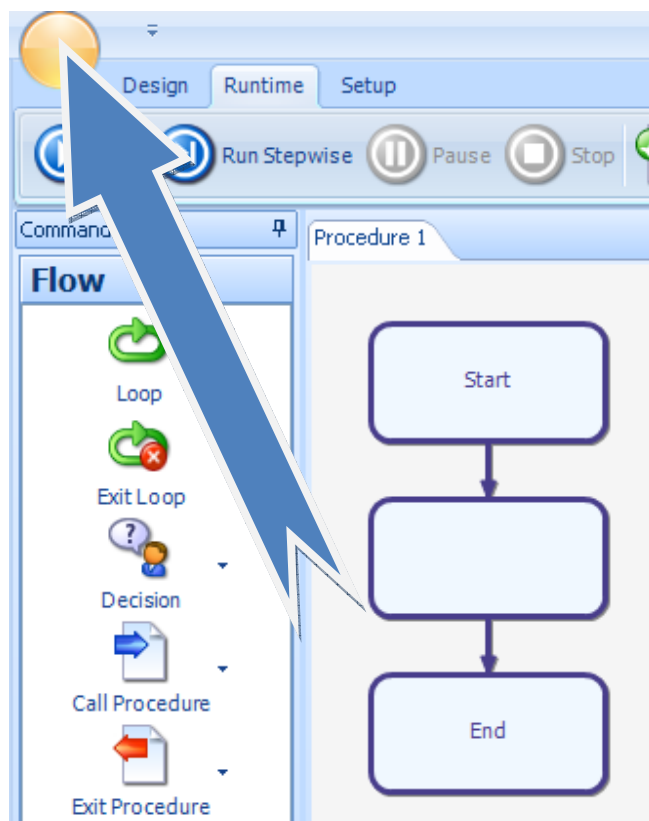
The user also has a number of Overlord3 environmental options, such as docking sub windows, hiding sub windows and colour options.

## 5.2 File ribbon options

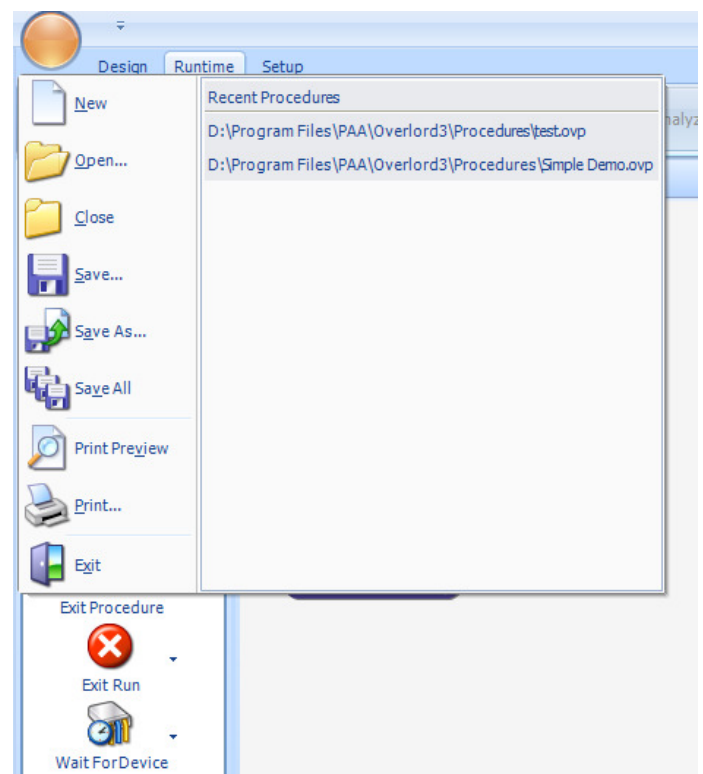
The top left of the Overlord3 screen, has two sections, the standard file manipulation options (Section 5.2.1 File options) and the ribbon options (Section 5.2.2 Ribbon options).

### 5.2.1 File options

To use the file options, select the blob in the top left of the Overlord3 main screen (File options selection). The file options menu then appears (File options menu).



**File options selection**



**File options menu**

The user can then perform one of the many standard file options:

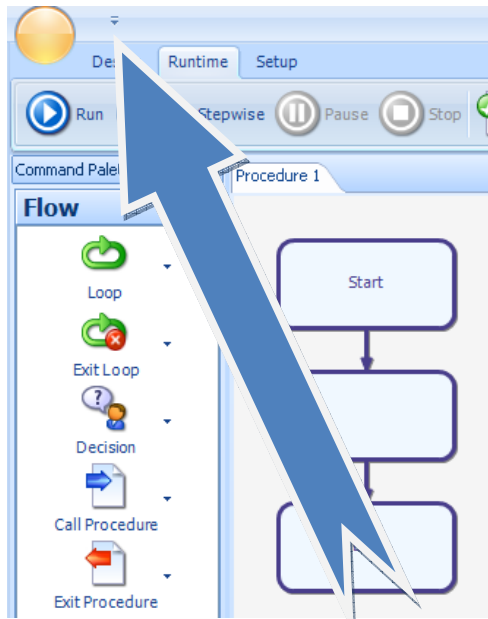
New	Open a new <b>Overlord3</b> procedure for editing
Open	Open an existing <b>Overlord3</b> procedure for editing
Close	Close the current active open <b>Overlord3</b> procedure
Save	Save the current active open <b>Overlord3</b> procedure with the same procedure name
Save As	Save the current active open <b>Overlord3</b> procedure with a different procedure name
Save All	Save all the current open <b>Overlord3</b> procedures
Print Preview	Throws up a dialog box showing how the procedure would look if printed
Print	Print the current active open <b>Overlord3</b> procedure
Exit	Exit

The user is also offered a list of recent opened **Overlord3** procedures on the right hand side of the panel.

## 5.2.2 Ribbon options

The **ribbon** options button is located next to the **File** options button (Ribbon Options, Figure 17 - Ribbon Options). To add quick access icons to the quick access ribbon, follow the instructions below.

The user can manipulate the **ribbon** options drop down box (Ribbon drop down list).



Ribbon Options

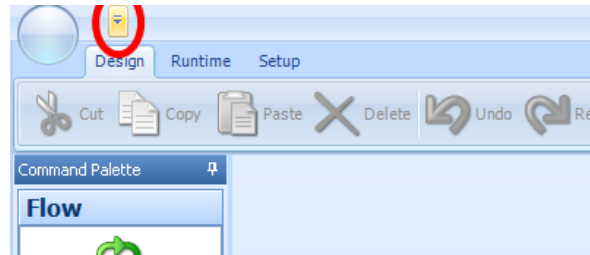
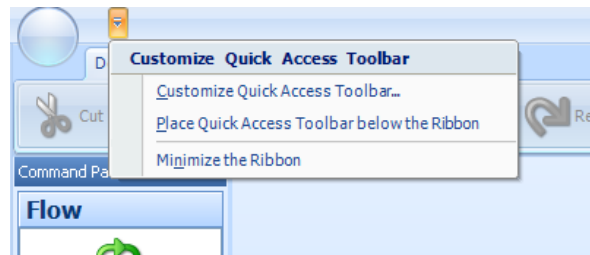


Figure 17 - Ribbon Options



Ribbon drop down list

### 5.2.2.1 Customize Quick Access toolbar

To **Customize Quick Access toolbar**, select that option on the drop down menu. This will bring up the configuration menu (Figure 18 - Toolbar configuration options). Select the **Run** option (Choose run toolbar to edit).

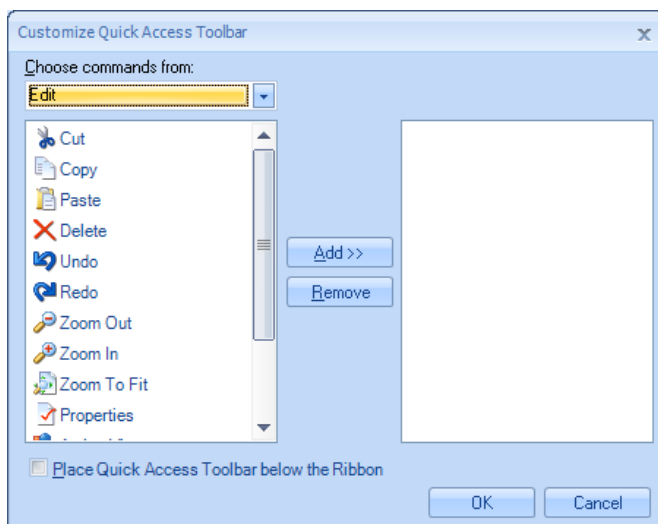
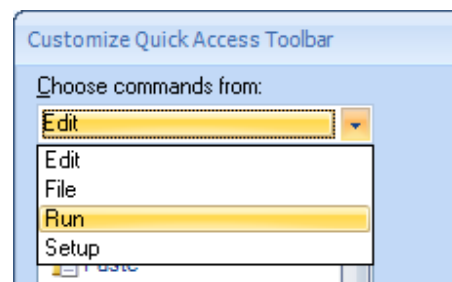
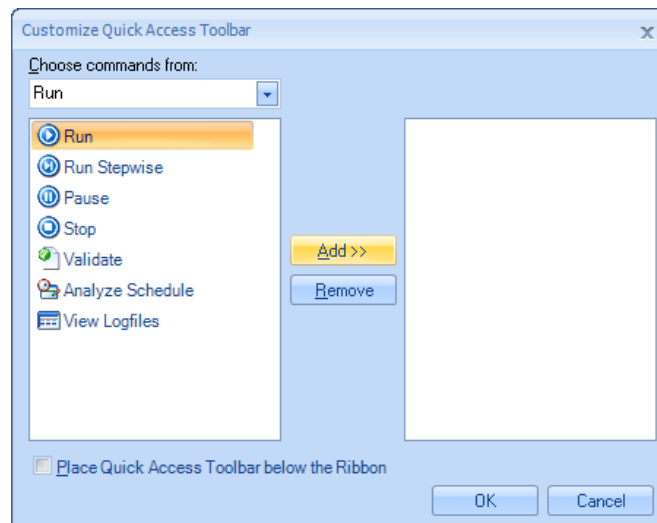


Figure 18 - Toolbar configuration options



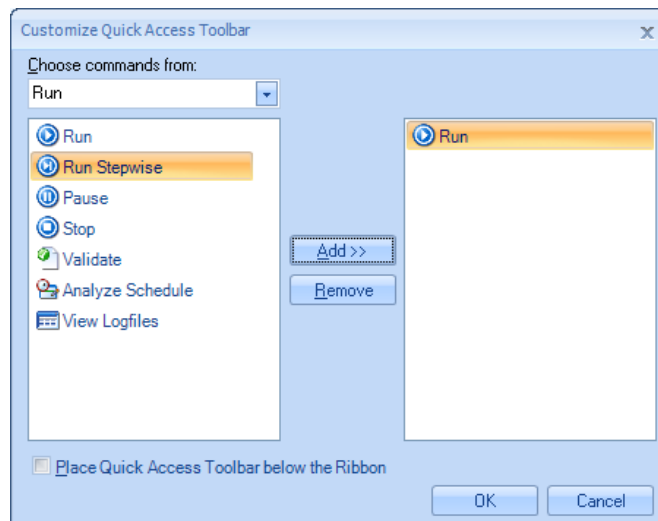
Choose run toolbar to edit

The options on the run menu then can be selected to be included in the quick access ribbon (Add icon to quick access).



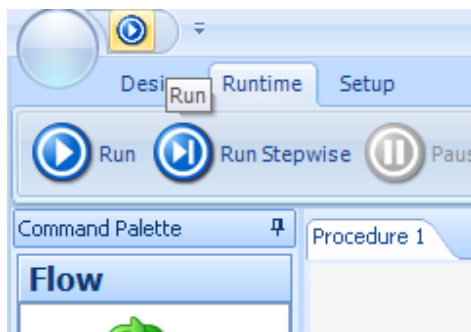
**Add icon to quick access**

Selecting **Add** moves the **Run** icon to the right hand pane (Run icon added).



**Run icon added**

Selecting **OK** will add the **Run** icon to the quick access ribbon (Run added to the quick access ribbon).



**Run added to the quick access ribbon**

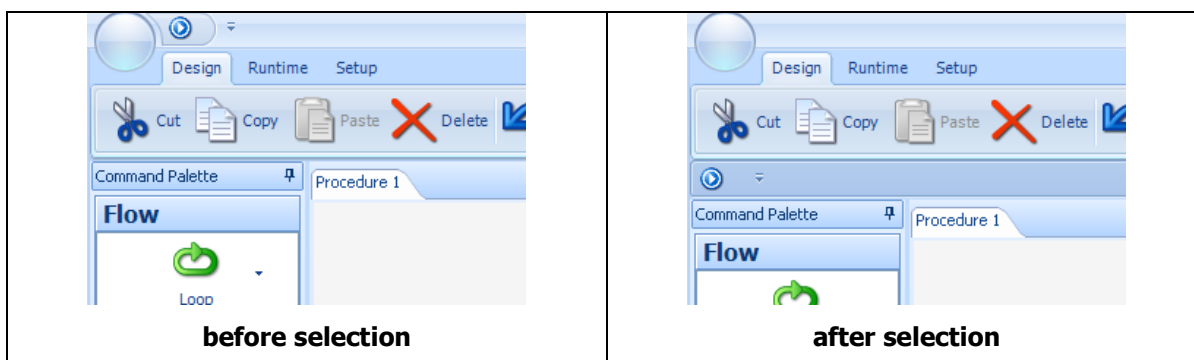
The **Run** option is available for easy access.

## ***5.2.2.2 Place the Quick Access Toolbar below the Ribbon***

Selection of this option allows the presentation of the options in a different order. The **Place the Quick Access Toolbar below the Ribbon** option is available from the menu (Ribbon drop down list).

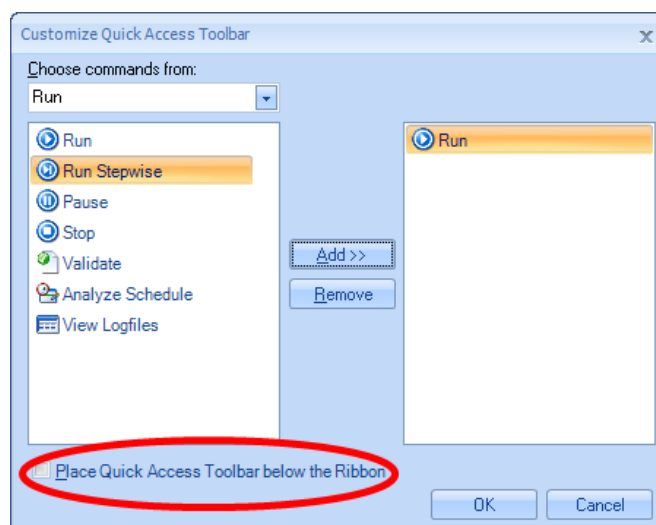
option is accessed from the drop down

Before the selection the top left of the screen is shown (before selection) after the selection the icons are reorganized (after selection).



This can be achieved in the editing dialog box (switch order in dialog box).





switch order in dialog box

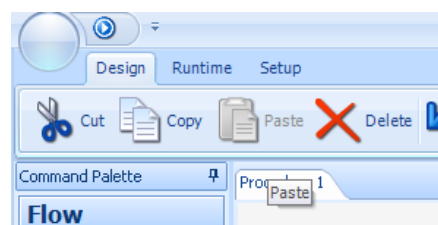
This can be reversed by clicking the option again and selecting **Place the Quick Access Toolbar below the Ribbon** from the menu, or deselecting the option in the dialog box (switch order in dialog box).

### 5.2.2.3 Minimize the Ribbon

The **Minimize the Ribbon** option (Ribbon drop down list) removes the ribbon completely (ribbon minimized). The ribbon can be restored by selecting **Maximize the Ribbon** from the same menu (ribbon maximized).



ribbon minimized

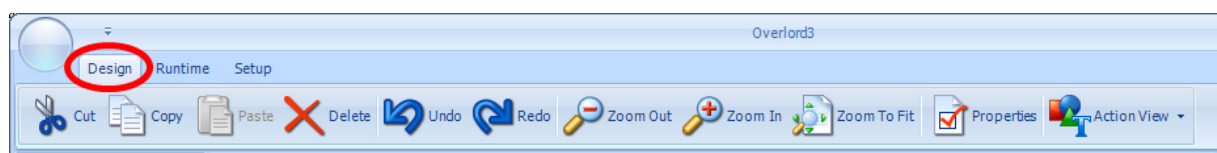


ribbon maximized

## 5.3 Top panel - tab and buttons

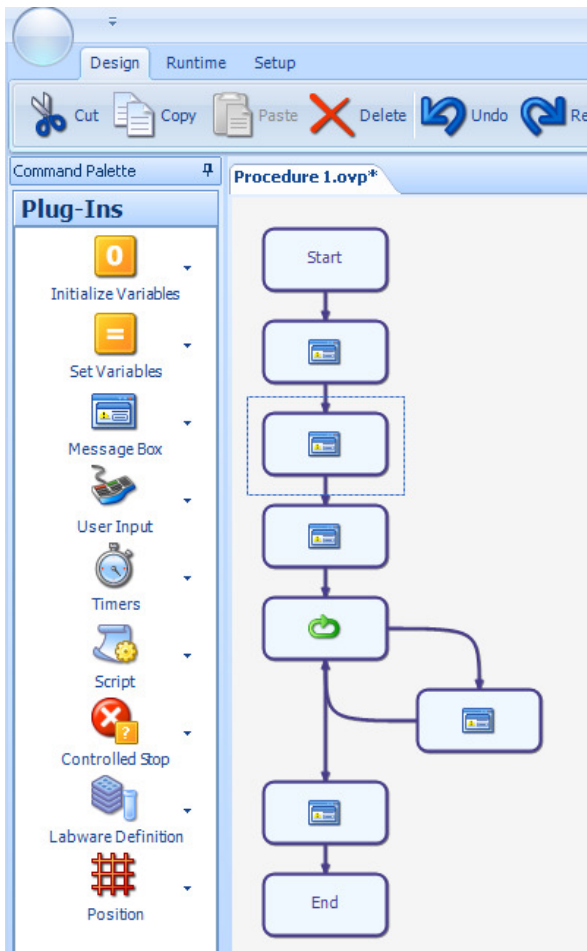
### 5.3.1 Design tab

The **Design** tab allows access to all the editing functions (Design tab for procedure editing).

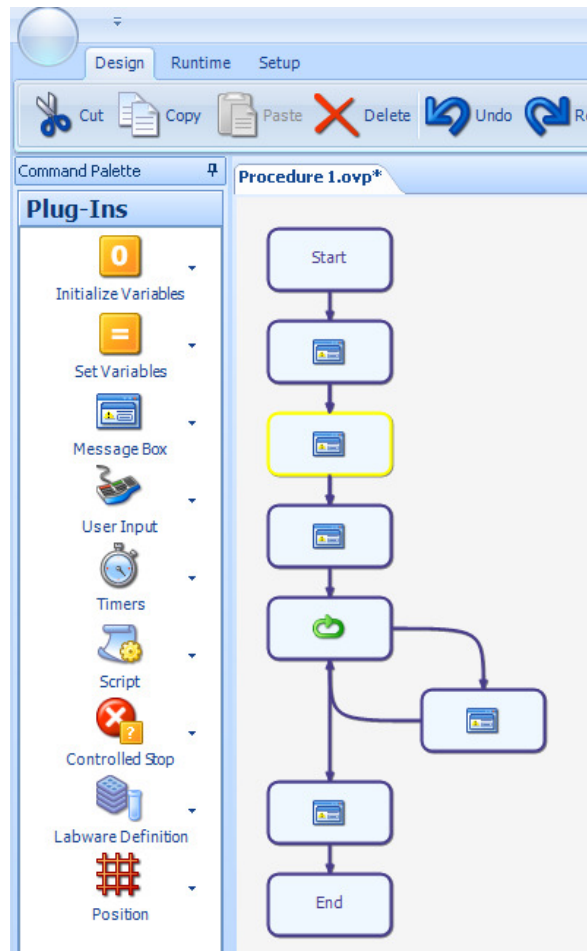


Design tab for procedure editing

The design panel offers similar functions to any editor. Selecting an action or actions in a procedure is completed by clicking the mouse left hand button in one position, keeping the mouse selected and bounding the action or actions, then releasing the mouse (bounding an Overlord3 action). The selected action or actions then appear with a yellow perimeter (the Overlord3 action is selected).



**bounding an Overlord3 action**



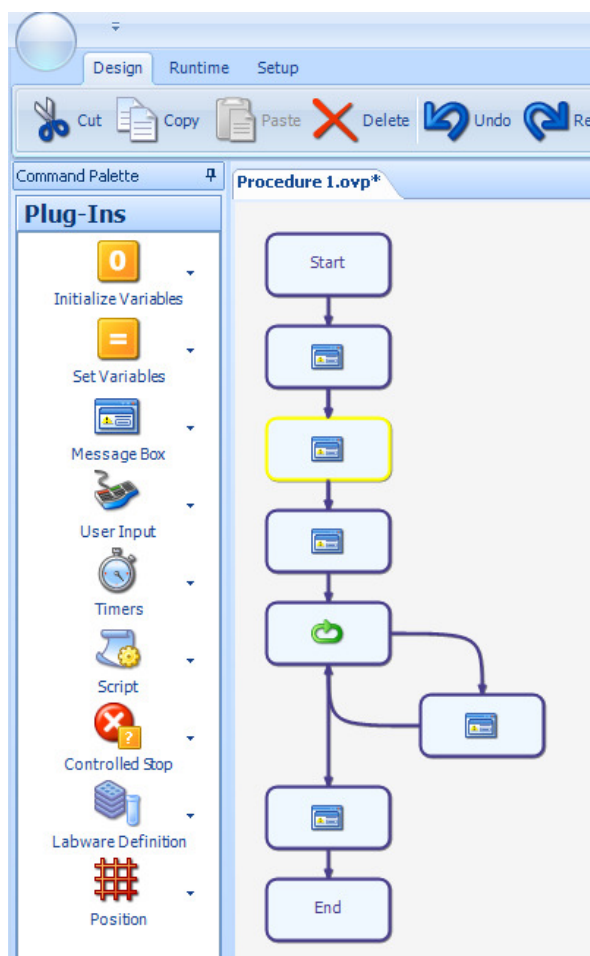
**the Overlord3 action is selected**

This action can then be manipulated with the other commands. Any of the icons may be greyed out as that function may not be available.

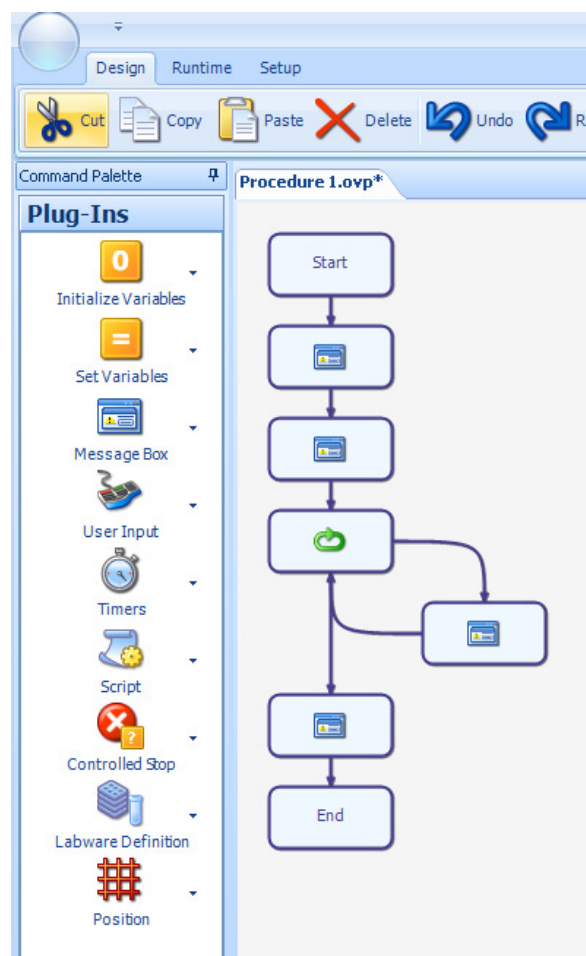


## 5.3.1.1 Cut

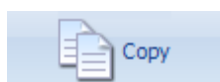
Cut a single or multiple actions in an **Overlord3** procedure. Taking a demonstration Overlord3 procedure, it is possible to remove an action by bounding the element (the Overlord3 action is selected) and selecting the **Cut** icon, the unwanted action will be removed (the action is removed) and stored for later use (Section 5.3.1.2 Copy135, Section 5.3.1.3 Paste).



the Overlord3 action is selected



the action is removed



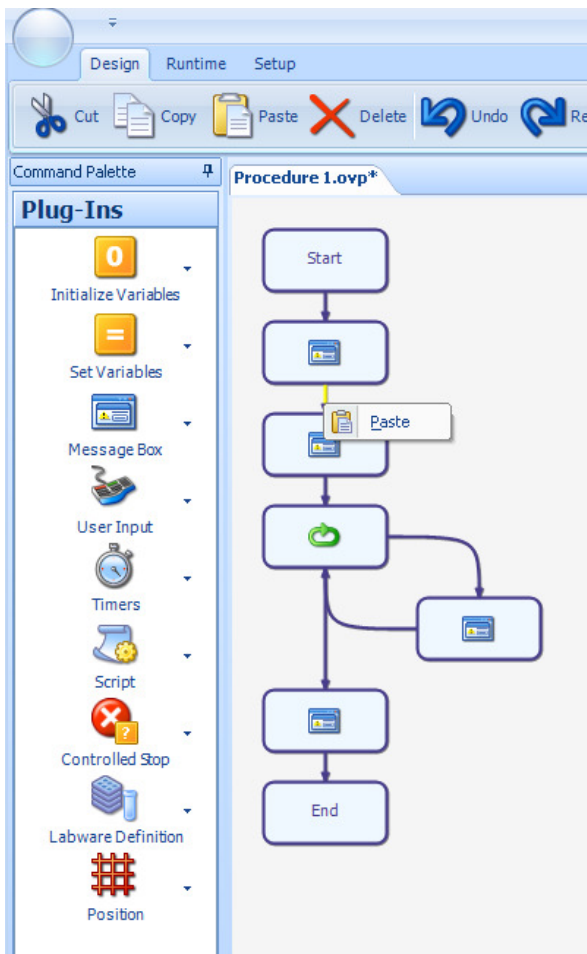
### 5.3.1.2 Copy

Copy a single or multiple actions in an **Overlord3** procedure. Select an Overlord3™ action (the Overlord3 action is selected) and select the **Copy** icon. The action is stored for later use and the copied action or actions can be pasted in this or another **Overlord3** procedure (Section 5.3.1.3 Paste). Control-C also works to copy an action or actions to the copy buffer.

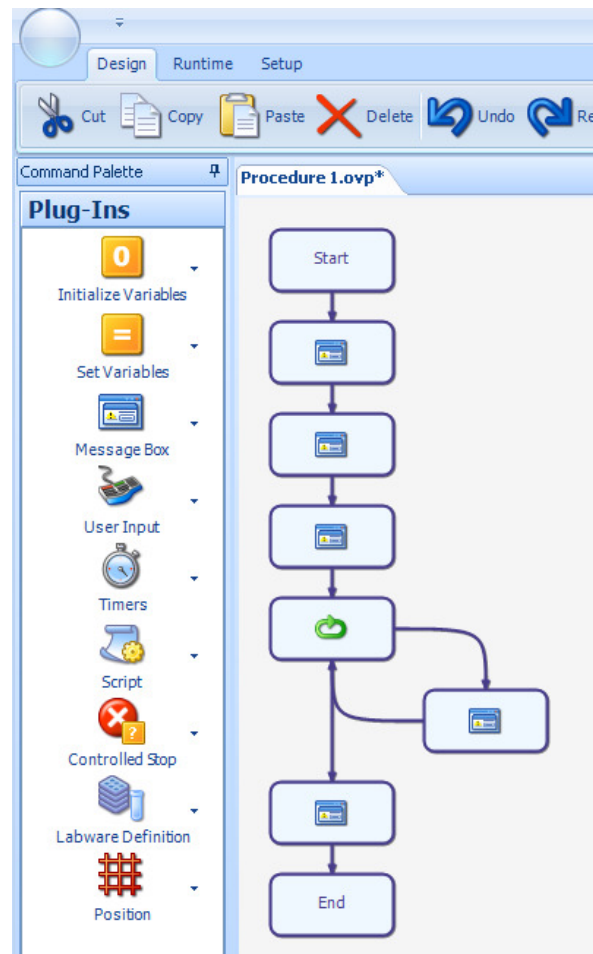


### 5.3.1.3 Paste

The paste icon only becomes available when an action has been copied or cut in a previous action. Select with the mouse right hand button the location where the action or actions are to be pasted, the link will turn yellow (select a position to paste an action). Click the **Paste** box, and the stored action will be pasted into that position (action pasted in selected position).

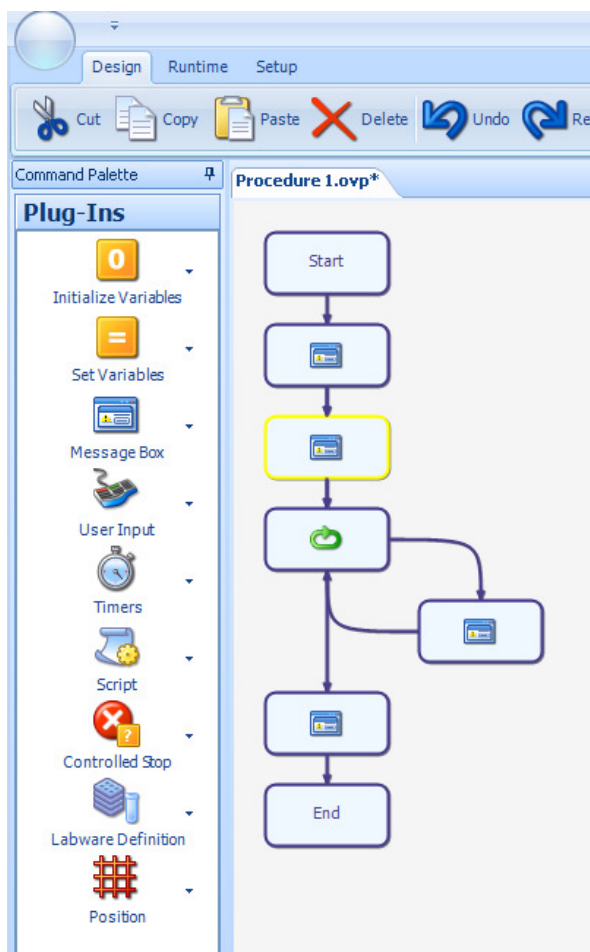


**select a position to paste an action**

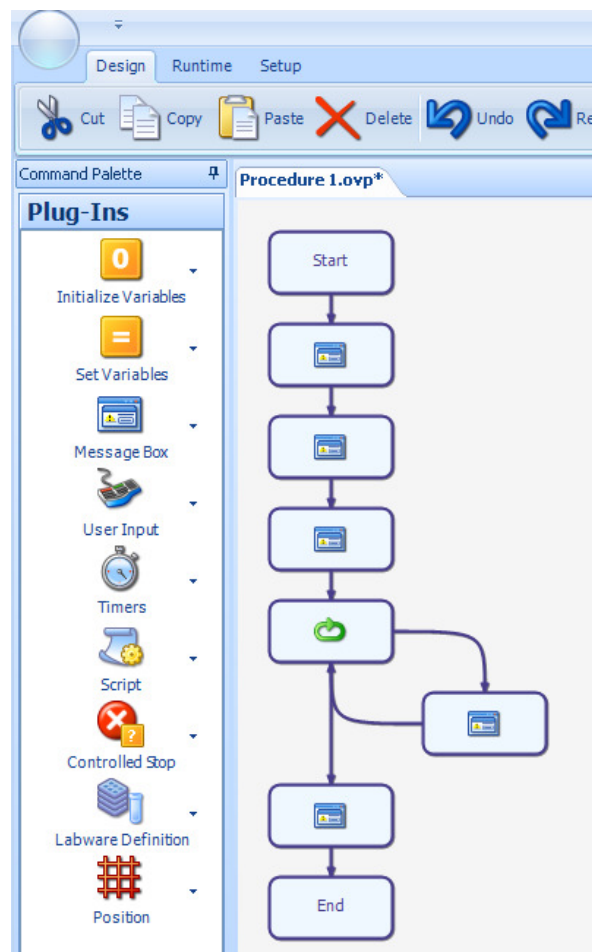


**action pasted in selected position**

The user can also select an action in a procedure (select a position to paste an action) and choose the Paste icon (action pasted in selected position). Control-V also works to paste the copy buffer to the required position in the procedure.



**select a position to paste an action**

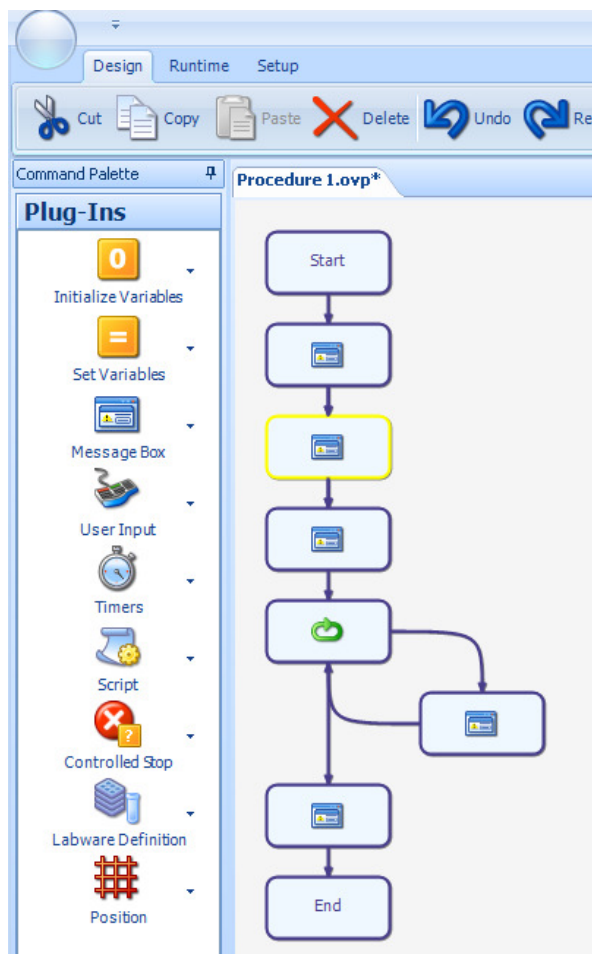


**action pasted in selected position**

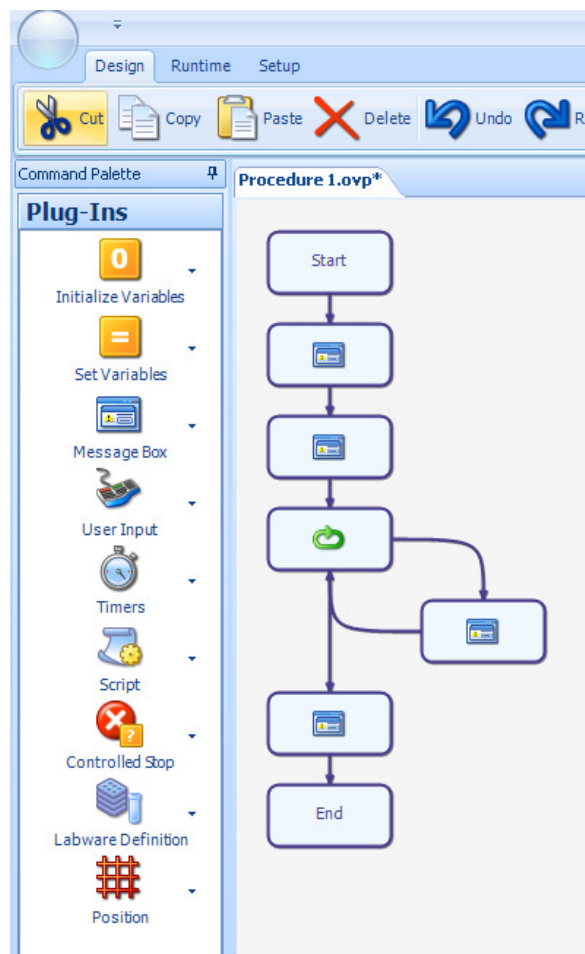


#### **5.3.1.4 Delete**

To delete an action in a procedure (the Overlord3 action is selected), select the action or actions to be deleted, and select Delete (the action is removed).



the Overlord3 action is selected



the action is removed



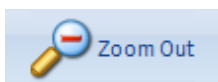
## 5.3.1.5 Undo

If the user wishes to undo the last action, just select the **Undo** action. This has multiple layers of undo.



## 5.3.1.6 Redo

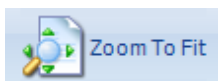
If the user uses undo too many times, just select the **Redo** action to retract the last item.



## ***5.3.1.7 Zoom Out Zoom In***

**Zoom Out** allows the user to see more of the procedure at the expense of clarity. See below for the effect of **Zoom Out** before (**Error! Reference source not found.**) and after (**Error! Reference source not found.**) selecting the **Zoom Out** button.

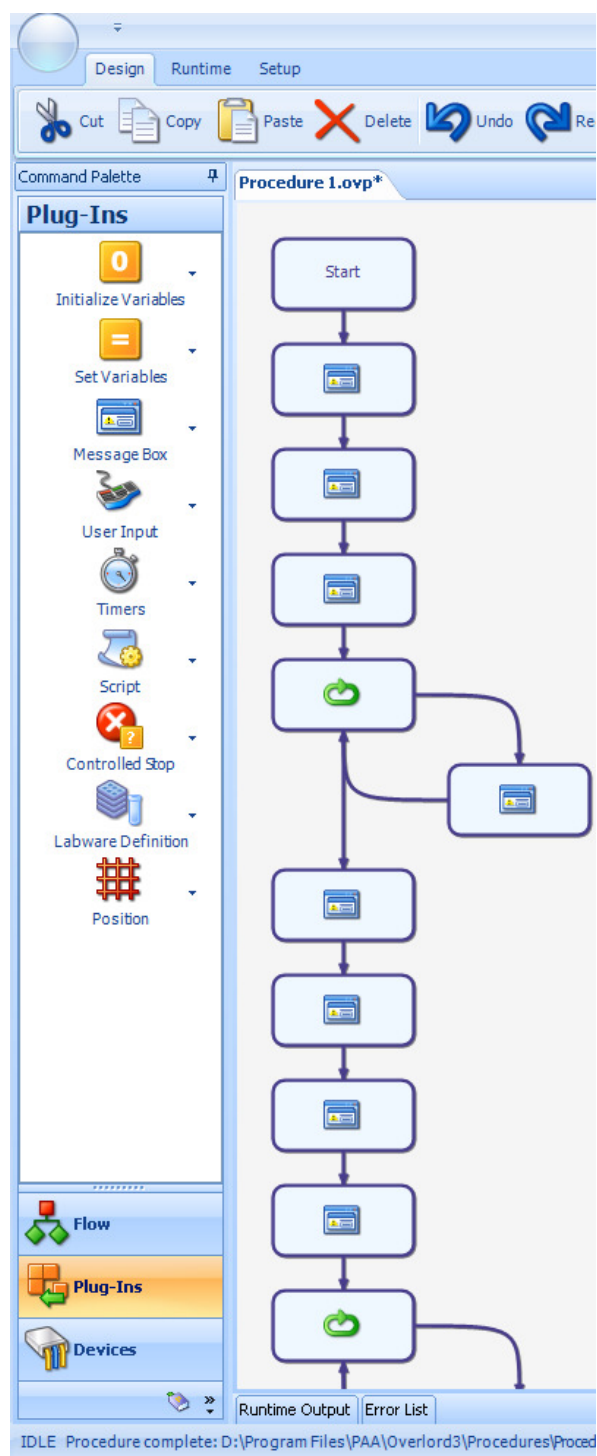
**Zoom In** allows the user to see less of the procedure at the expense of quantity of information. See below for the effect of **Zoom In** before (**Error! Reference source not found.**) and after (**Error! Reference source not found.**) selecting the **Zoom In** button.



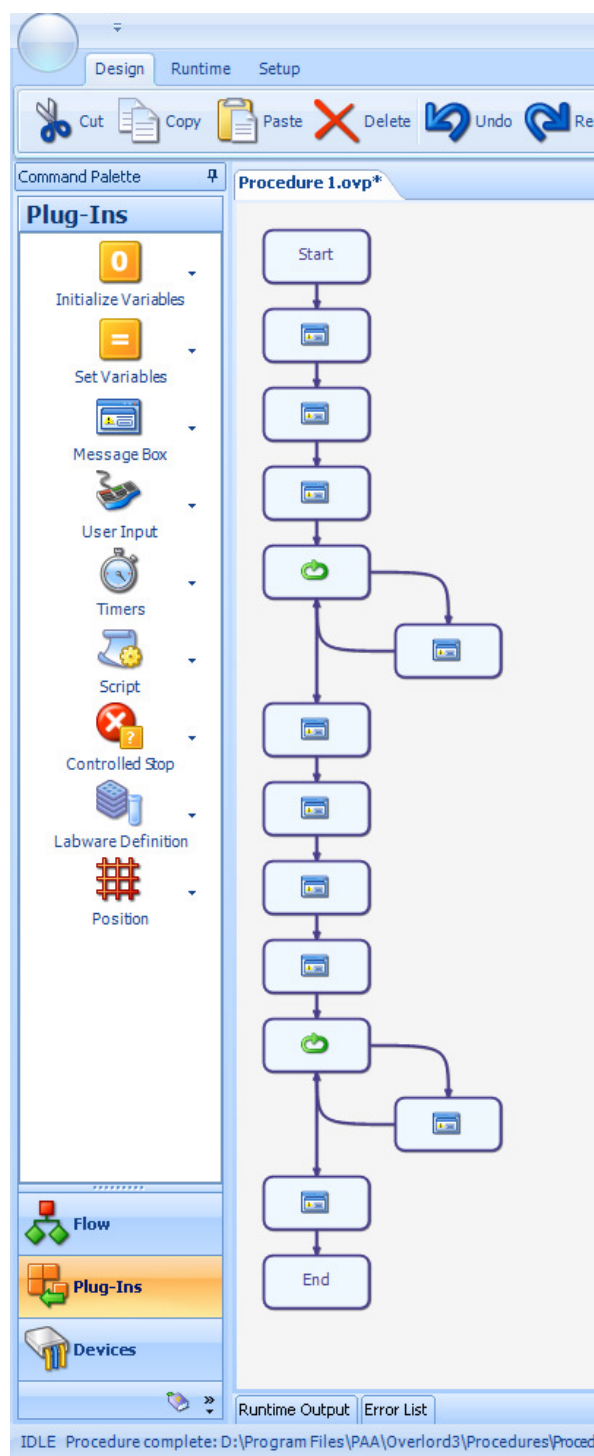
## ***5.3.1.8 Zoom to Fit***

The user can get the whole procedure on one screen at any time using the **Zoom To Fit** button. This is at the expense of procedure details. See below for the effect of **Zoom To Fit** before (before Zoom To Fit) and after (after Zoom To Fit) selecting the **Zoom To Fit** button.





**before Zoom To Fit**



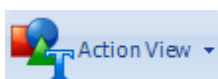
**after Zoom To Fit**





### **5.3.1.9 Properties**

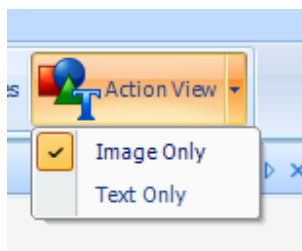
The **Properties** button allows access to the details on any **Overlord3** procedure step (the edit screen). This is the same as double clicking on the procedure step.



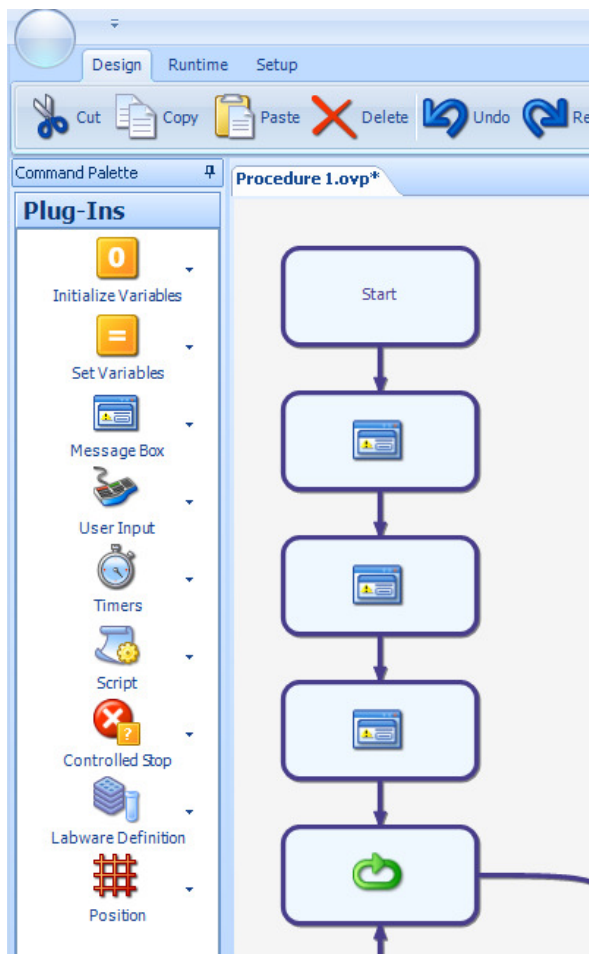
### **5.3.1.10 Action View**

Action View allows the user to see the comments relating to the **Overlord3™** procedure step rather than the graphic representation. Selecting the Action View command, allows the user to choose in a menu between Graphic Only and Text Only (Action View Menu).

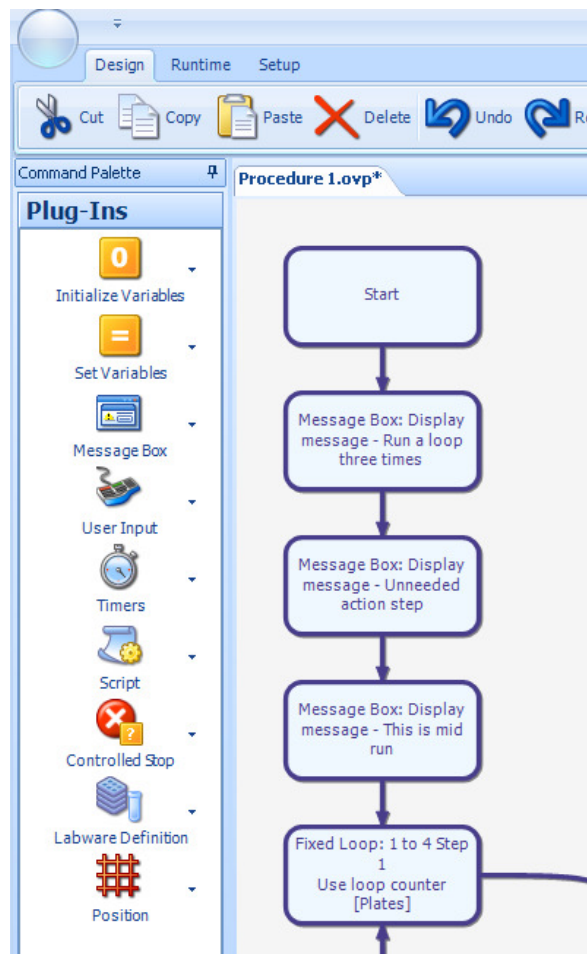
The graphic only (Action View, Image Only) can be compared with text only (Action View, Text Only) below.



## Action View Menu



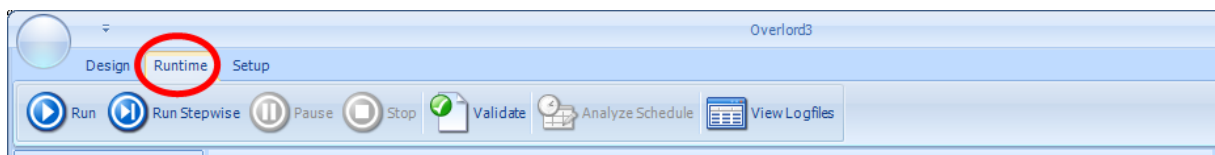
**Action View, Image Only**



**Action View, Text Only**

### 5.3.2 Runtime tab

The **Runtime** tab allows access to all the runtime functions (Runtime tab for runtime control).



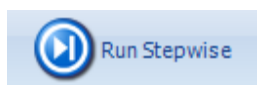
#### Runtime tab for runtime control

The Runtime panel offers similar functions to any DVD player. When the user has completed the procedures for the required automation method, the procedure is ready to run.



#### 5.3.2.1 Run

Run the current procedure.



#### 5.3.2.2 Run Stepwise

Run the current procedure step-by-step. After each procedure action has been completed, **Overlord3** will expect the **Run Stepwise** icon to be pressed. **Run** continuously or **Stop** are also active for use at each step of the procedure.



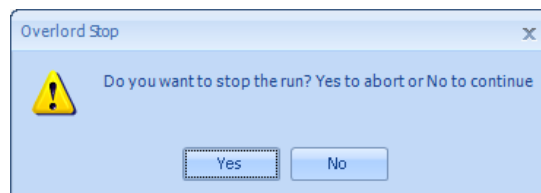
#### 5.3.2.3 Pause

Pause the current running procedure. **Run** continuously, **Run Stepwise** or **Stop** are possible options after pausing the run.



#### 5.3.2.4 Stop

Stop the current running procedure. The user will be presented with a dialog box (Stop the run dialog box) to confirm the cessation of the run (select **Yes**) or to continue from the stopping point (select **No**).

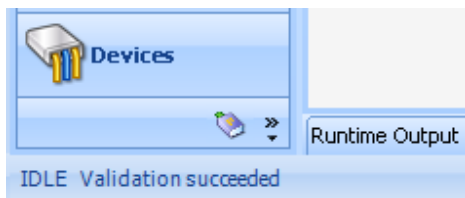


**Stop the run dialog box**

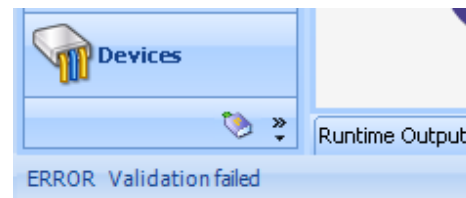


### 5.3.2.5 Validate

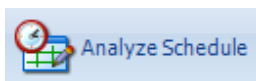
Validate the commands in the current procedure. The bottom left of the screen will show two messages, either **Validation succeeded** (Validation succeeded) or **Validation failed** (Validation failed).



**Validation succeeded**

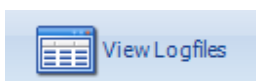


**Validation failed**



### 5.3.2.6 Analyze Schedule

Analyses the current Scheduled Procedure file to show the estimated order of actions at runtime. This includes estimated duration for the run.

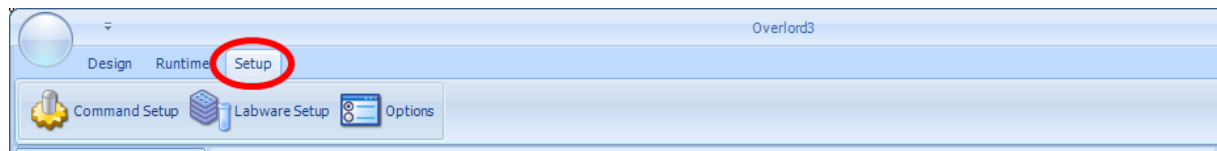


### 5.3.2.7 View Logfiles

View the runtime log files. This is explained in more detail in a later section.

## 5.3.3 Setup tab

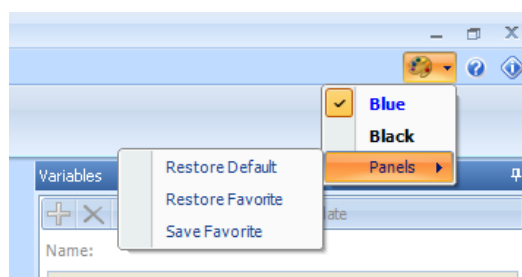
The **Setup** tab allows access to all the environment functions (Runtime tab for runtime control).



**Setup tab for system configuration**

## 5.3.4 Environmental controls, main additional options

The main options for the environment control are found in the artist palette icon (Main environment control).

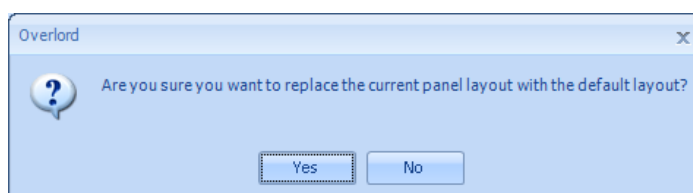


**Main environment control**

This options allow selection of the color scheme.

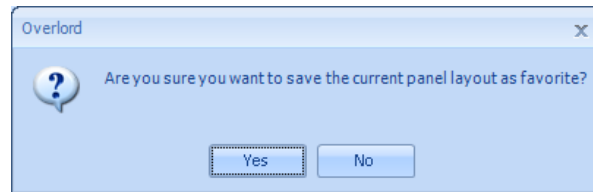
The user can have either a favorite environment configuration or the default configuration.

If the user has reconfigured the environment and wishes to return to the factory settings, they should select the **Restore Default** option and answer **Yes** to the dialog box ( restore default environment).



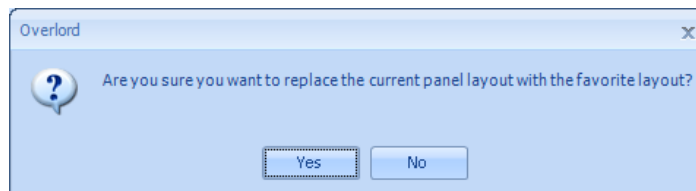
**restore default environment**

To save a favorite environment, select **Save Favorite**, and answer **Yes** to the dialog box (save environment as favorite)



**save environment as favorite**

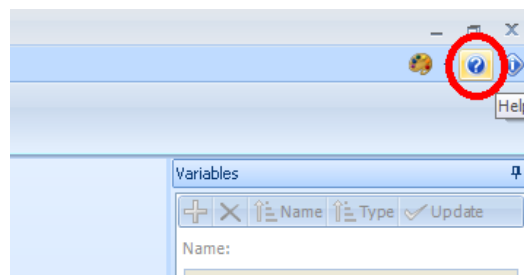
To restore a favorite environment, select **Restore Favorite**, and answer **Yes** to the dialog box ( restore favorite environment)



**restore favorite environment**

## 5.3.5 Environmental controls, help

The user has access to **Help** at any time with the help icon (Help).



**Help**

This shows the **Help** dialog box (**Error! Reference source not found.**) with standard access to the **Contents**, **Index** and **Search** functions.

## 5.3.6 Environmental controls, about

The current version of Overlord3 can be determined from the **About** dialog box. The **About** dialog box is displayed with version number and copyright information.

## 5.4 Right panel

### 5.4.1 Variables

There are 4 types of variable:

**Text**, this is data that is alphanumeric, it is used to handle textual information.

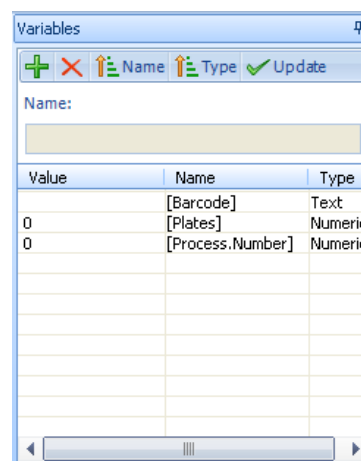
**Numeric**, this is data that has number information only (100, 10, 1, 0.1, 0.01 are all numbers)

**TrueFalse**, this is also known as a Boolean operator with values of True and False, this is used to describe the state of an object, e.g. the plate is loaded, True or False

**Object**, this is a more complicated structure for programmers to use in scripts, in order to use an Object created in one script in another. The Object would typically be a C# or VB.NET Object, for example a System.Collections.Generic.List(Of String) Object to store a list of barcode values.

The user should be aware that the number 1 and the text value 1 are completely different entities and should be treated differently by the user and are treated differently by Overlord3. The numeric value 1 can be compared to other numbers such as 0.1 or 10. The text value 1 is a statement of information only.

The variable panel (Main screen - Variables) lists the variables that are available to the user in the current Overlord3™ environment.



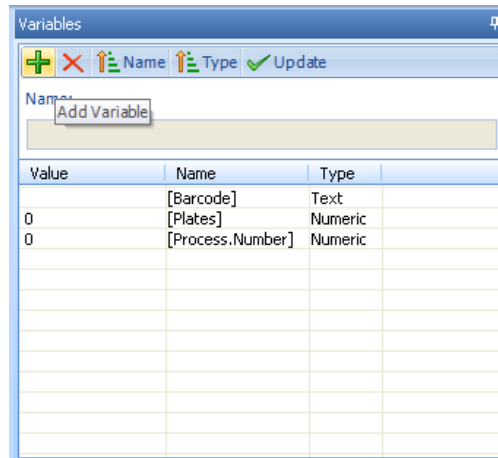
Value	Name	Type
0	[Barcode]	Text
0	[Plates]	Numeric
0	[Process.Number]	Numeric

**Main screen - Variables**

To add a new variable select the green plus



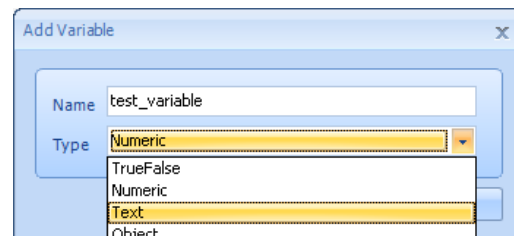
(add a variable icon).



**add a variable icon**

The user should see a dialog box to enter the new variable name. The user should enter the new variable name, in this example test\_variable and choose the variable type from the drop down list, in this example Text

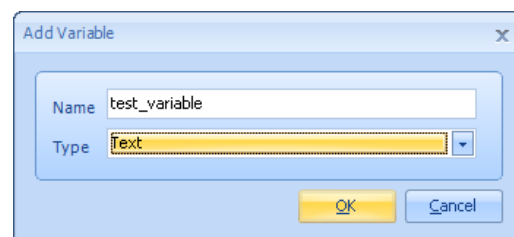
(add variable dialog box)



**add variable dialog box**

The user can then select OK to continue

(continue to add a variable).



**continue to add a variable**



This will update the Overlord3 main screen variable list

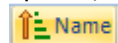
(updated variable list on main screen).

The screenshot shows the 'Variables' window with the 'Update' button (a green checkmark icon) highlighted. The window contains a table with the following data:

Value	Name	Type
	[Barcode]	Text
0	[Plates]	Numeric
0	[Process.Number]	Numeric
	[test_variable]	Text

**updated variable list on main screen**

When the list has a large number of entries, it is useful to sort the list, there are two options, by name using the Name button



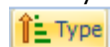
(variables, sort by name).

The screenshot shows the 'Variables' window with the 'Sort By Name' button highlighted. The window contains a table with the following data:

Value	Name	Type
	[Barcode]	Text
0	[Plates]	Numeric
0	[Process.Number]	Numeric
	[test_variable]	Text

**variables, sort by name**

And by variable type with the Type button



(variables, sort by type).

The screenshot shows the 'Variables' window with the 'Type' button (a green checkmark icon) highlighted. The window contains a table with the following data:

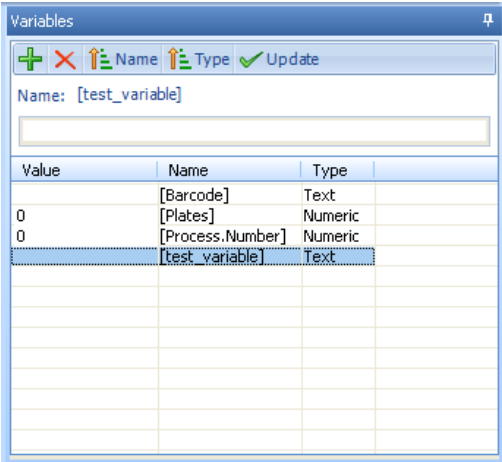
Value	Name	Type
0	[Plates]	Numeric
0	[Process.Number]	Numeric
	[Barcode]	Text
	[test_variable]	Text

**variables, sort by type**


Once Overlord3™ has been started the variable values persist until changed by the system. For example, it is necessary to initialise the variables at the beginning of a run to ensure that all the numbers are initialised to zero, and all the strings are set to no characters.

To change the value of a variable manually, select the variable of which you wish to change the value.

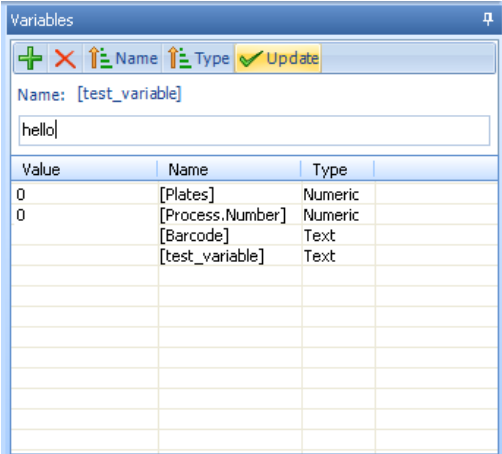
(changing the value of a variable)



**changing the value of a variable**

Enter the new value under the Name text entry box, and select Update .

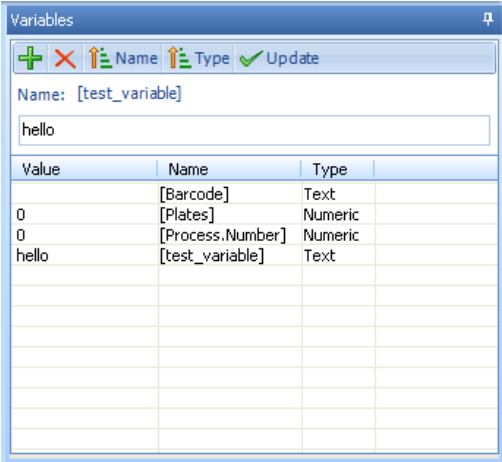
(variable value changes).




**variable value changes**

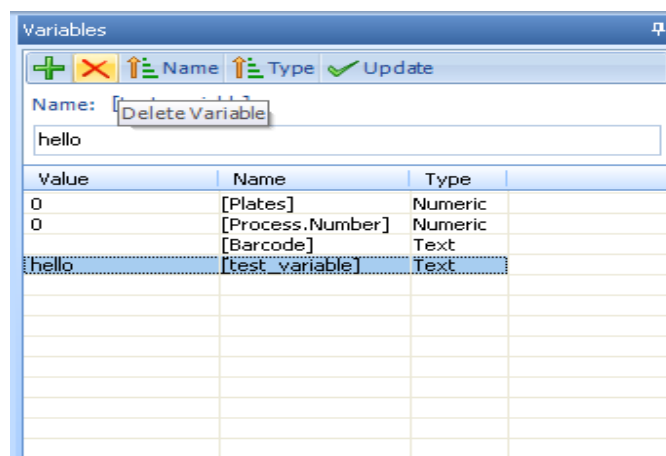
The variable value will now be set to the new value.

(new variable value)



**new variable value**

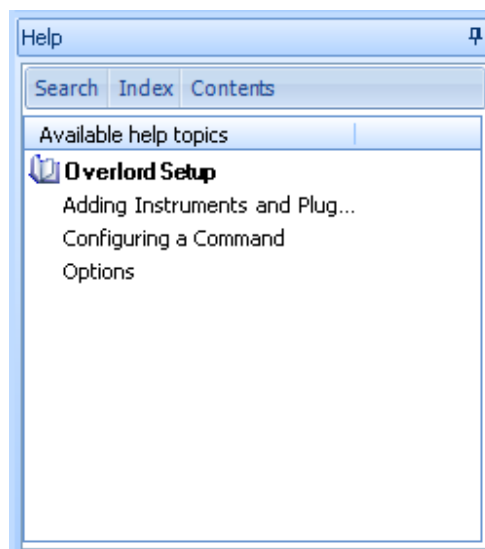
To delete a variable select the variable name and then click the red cross .  
(delete selected variable)



**delete selected variable**

## 5.4.2 Help

The context sensitive **Help** is available for most areas of the screen, command and runtime information. The help screen is updated depending on the screen area last selected.



**Main screen - help**

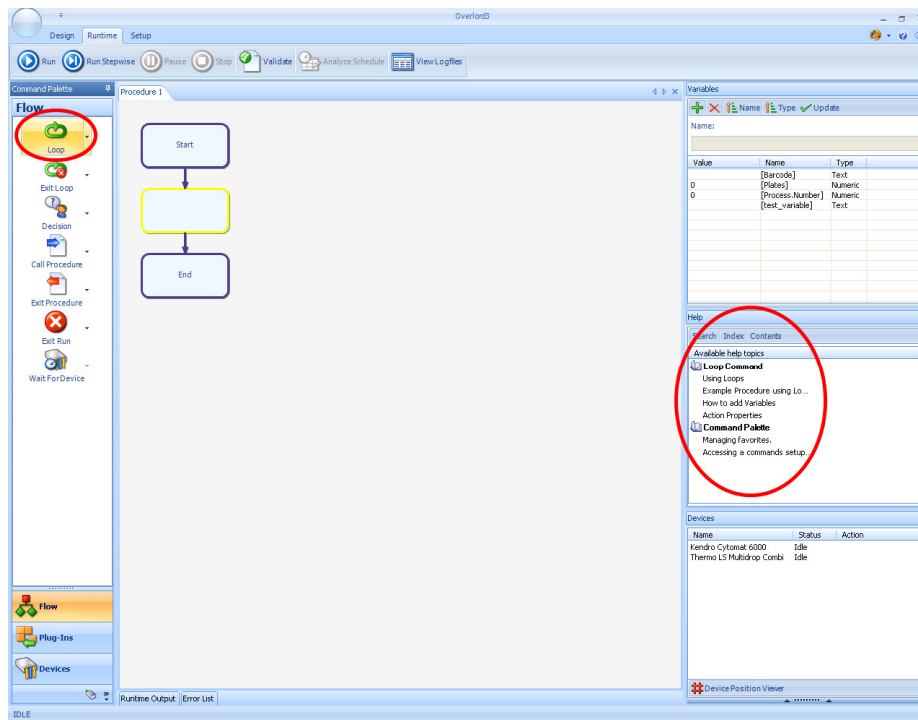
If another section of the screen is chosen, for example the **Loop** command in the **Flow** command area, the help box information changes (Help panel, Loop command).

Selecting one of the information lines in the **Help** area will bring up an information dialog box (Using the help panel for the **Loop** command).

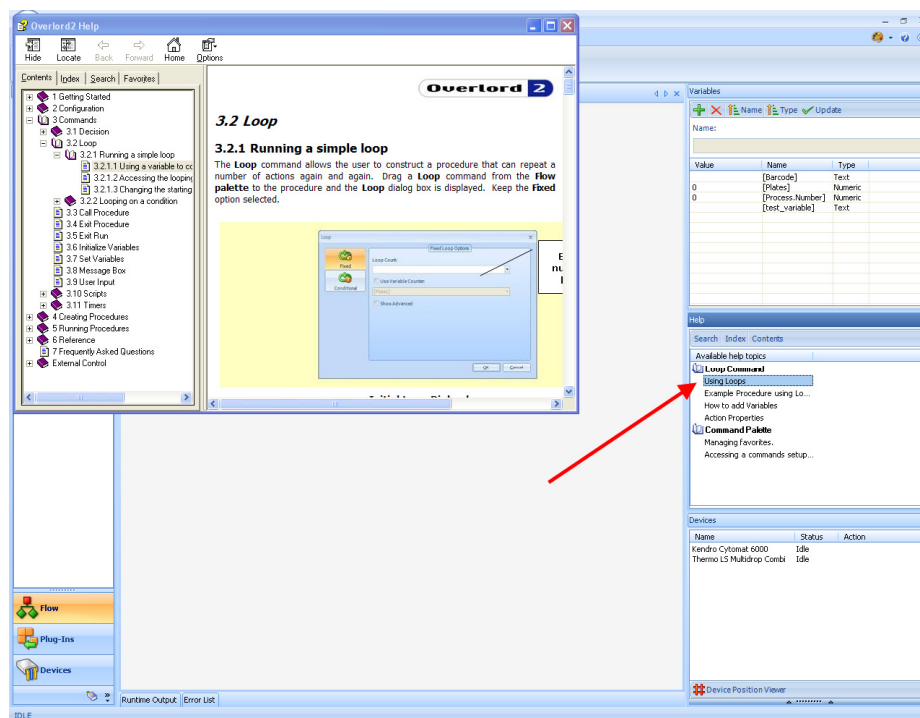
Searching for information on a set subject, chose the **Search** tab from the **Help** area. The standard keyword search panel is displayed (Searching the Help file).

To search for a keyword, select the **Index** tab of the **Help** area.(Searching the keyword index).

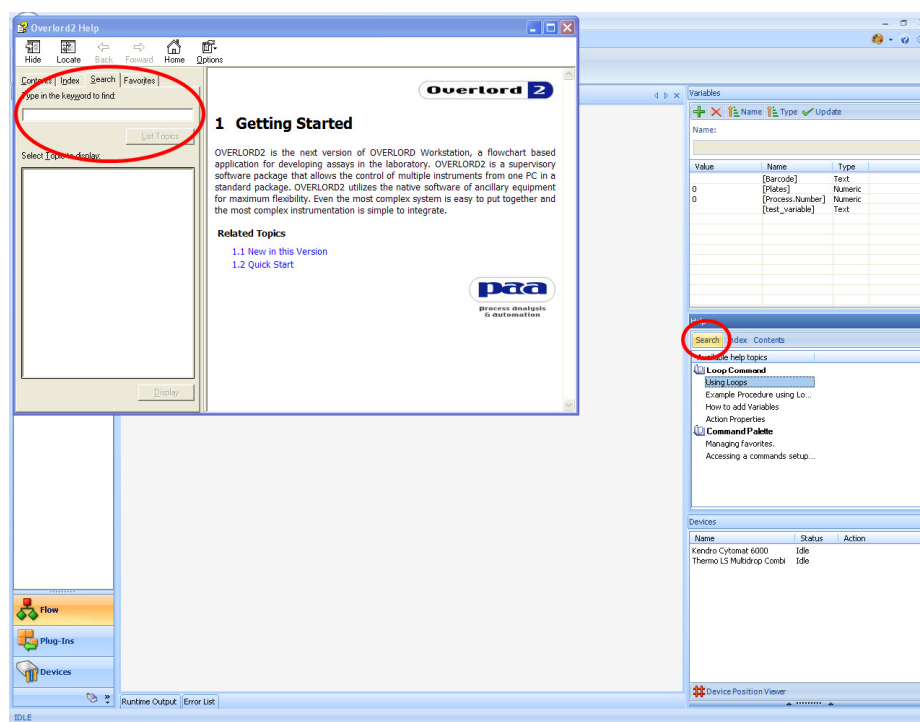
To view the **Contents** of the help file, select the **Contents** tab (Searching the Contents).



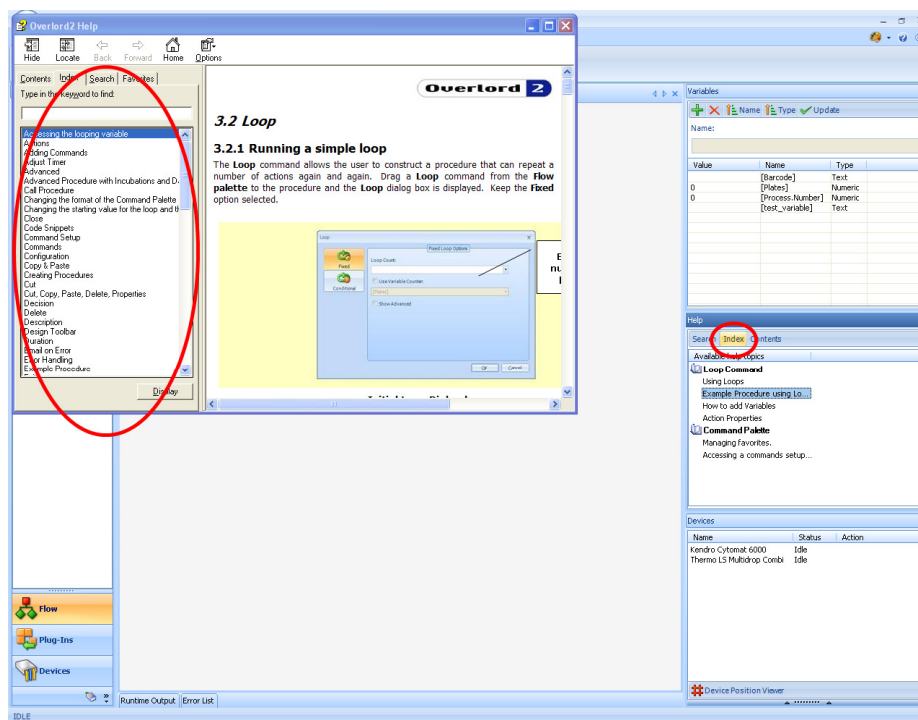
**Help panel, Loop command**



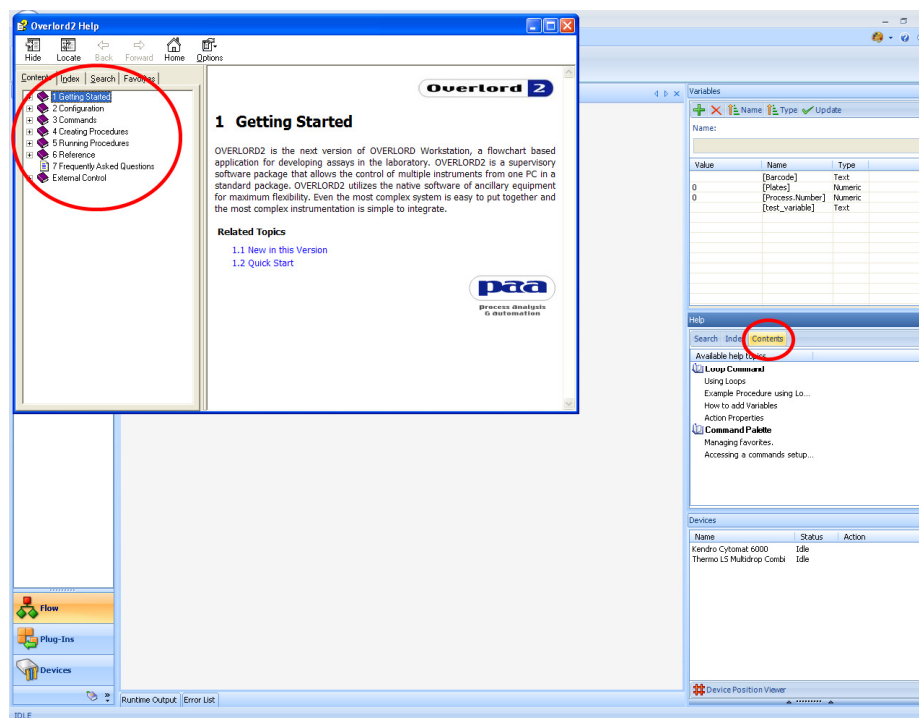
**Using the help panel for the Loop command**



**Searching the Help file**



**Searching the keyword index**



## Searching the Contents

### 5.4.3 Devices

The devices area of the main screen displays the currently installed instruments/devices on the system (Main screen - devices). To add, remove or change this list, move to the section relating to setup.

This information box acts as a runtime information box, showing if the instruments are **Idle** (Main screen - devices), **Busy** (Devices - Busy attribute) or in an **Error** status. The instrument is highlighted red on error.

Devices		
Name	Status	Action
Kendro Cytomat 6000	Idle	
Thermo LS Multidrop Combi	Idle	

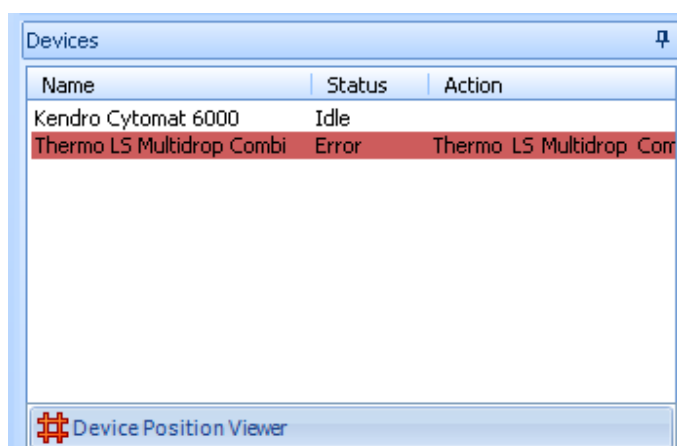
# Device Position Viewer

Main screen - devices

Devices		
Name	Status	Action
Kendro Cytomat 6000	Idle	
Thermo LS Multidrop Combi	Busy	Thermo LS Multidrop Com

# Device Position Viewer

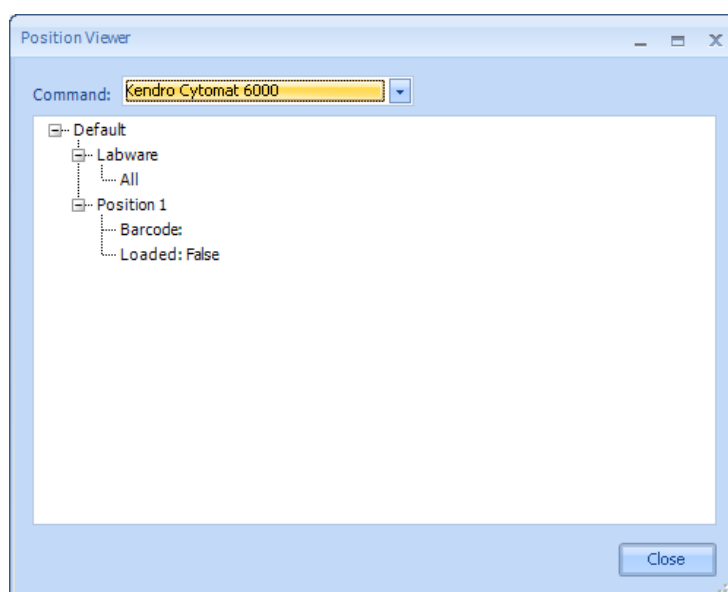
Devices - Busy attribute



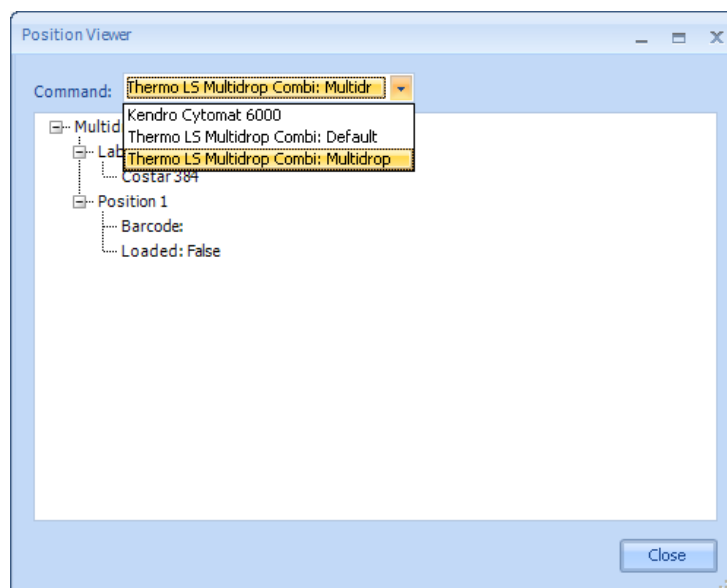
Name	Status	Action
Kendro Cytomat 6000	Idle	
Thermo LS Multidrop Combi	Error	Thermo LS Multidrop Combi

**Devices - Error attribute**

The **Device Position Viewer** allows the user to see the current status of all the equipment on the system, and the plate load status. Select **#Device Position Viewer** to see the information, choosing the drop down box (Position Viewer, change instrument) to change instrument.

**Position Viewer dialog box**

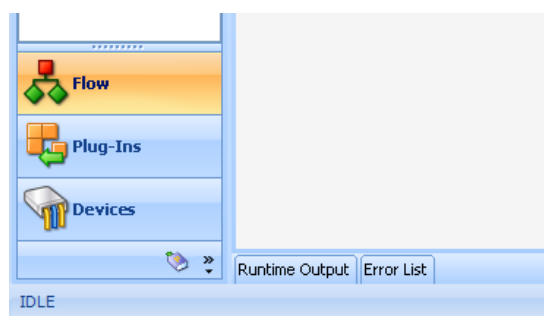




**Position Viewer, change instrument**

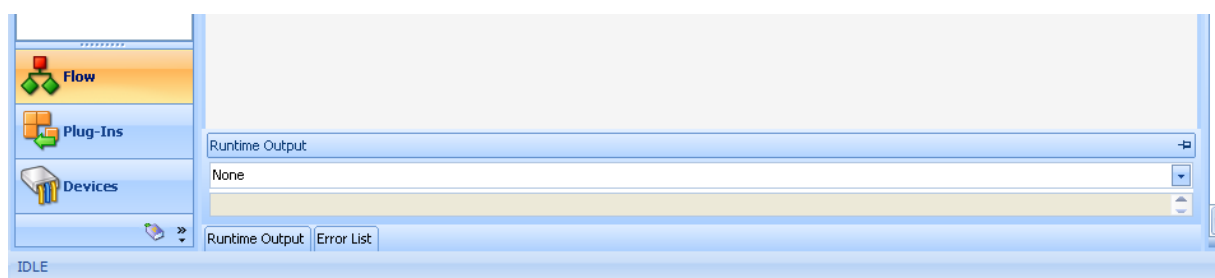
## ***5.5 Runtime information/Error list***

The runtime information box and runtime error list are stored as a tabs by default (Runtime Output).



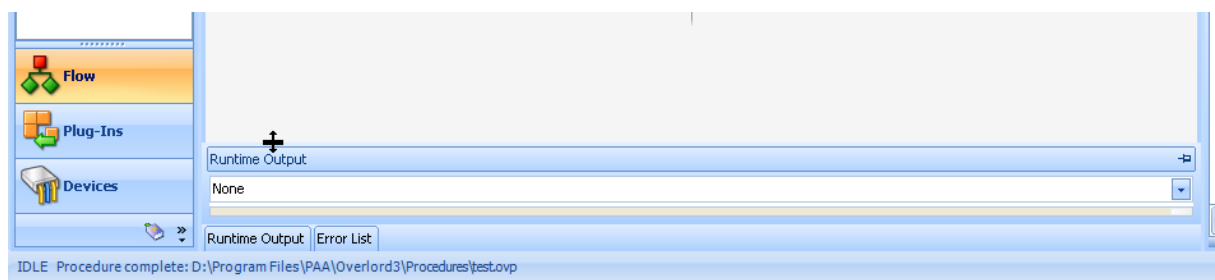
**Runtime Output**

Hover over or the tab to reveal the contents (Runtime Output tab expanded).



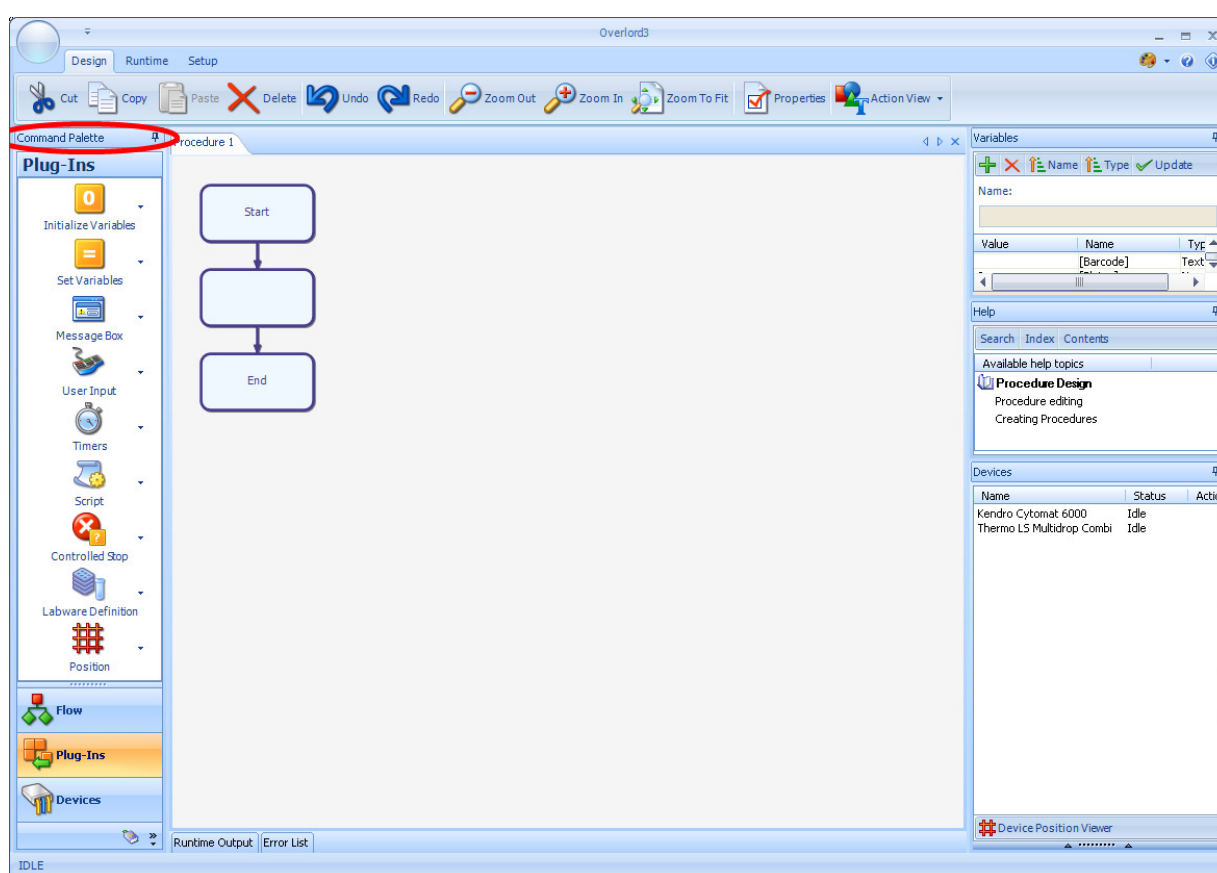
**Runtime Output tab expanded**

To view more of the panel, expand with the stretch tool (Runtime output resizing box),



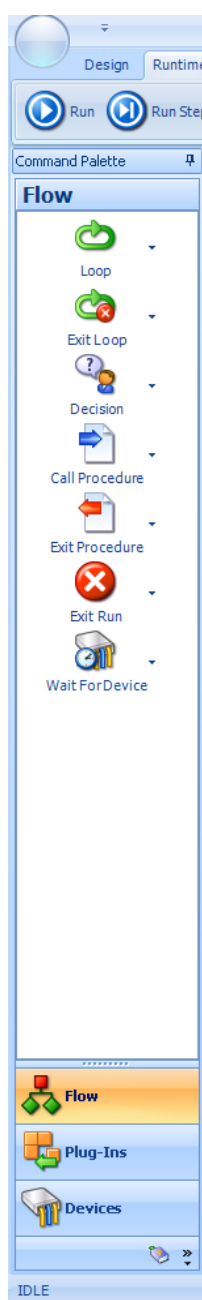
**Runtime output resizing box**

## 5.6 Command palette

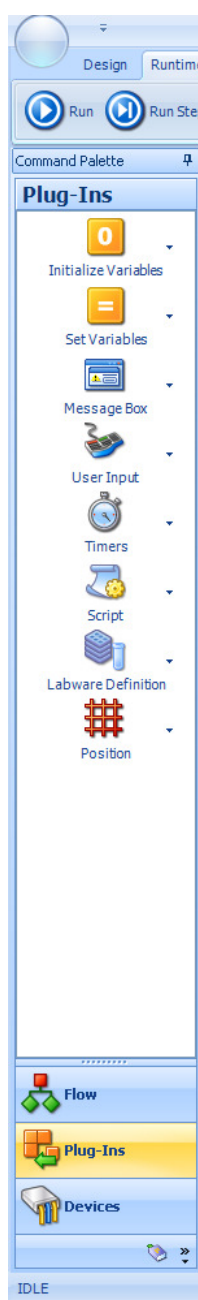


**Command palette**

These commands are listed into three different categories, **Flow**, **Plug-ins** and **Devices**. The start default is Flow (Figure 1.4 - Flow palette). Clicking on the **Plug-Ins** icon on the Command Palette will show the available Plug-In commands (Figure 1.5 - Plug-Ins palette). If you wish to see and used the installed instruments, select the **Devices** palette (Figure 1.6 - Devices palette).



**Flow palette**



**Plug-Ins palette**



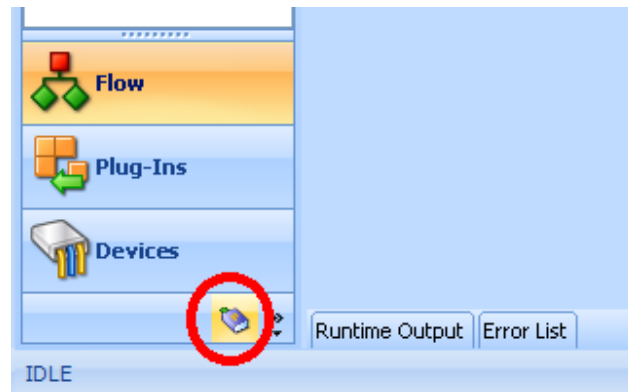
**Devices palette**

Each command can be dragged onto the **Procedure** flowchart to generate an **Action**, which uses the Command in a particular way. The follow section describes all of the common commands available and how they would be used within a **Procedure**.

These functions are explained in the Commands Section 4.

## 5.7 Favorites area

Selecting the favorites option (View Favorites palette option), allows a user to move frequently used commands into one area.

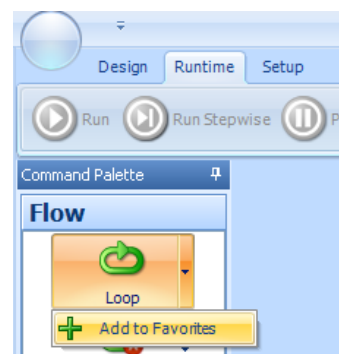


**View Favorites palette option**

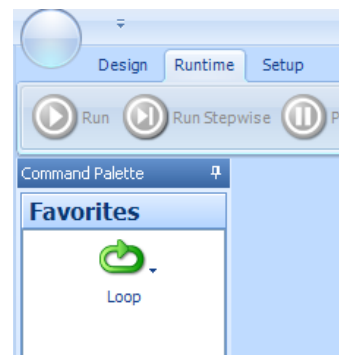
The Overlord **Favorites** palette, is initially empty (Favorites empty), however selecting an icon in another palette (e.g. **Loop** in **Flow**) and using the mouse in the drop down arrow on the right hand side of the **Loop** icon to bring up the drop down menu to select **Add to Favorites** (Loop, add to Favorites), allows the user to add a new option to the **Favorites** palette (Loop added to Favorites). This **Favorites** option can be removed by using the mouse in the drop down arrow on the right hand side of the **Loop** icon to bring up the drop down menu (Remove Loop from Favorites) and selecting **Remove from Favorites**.



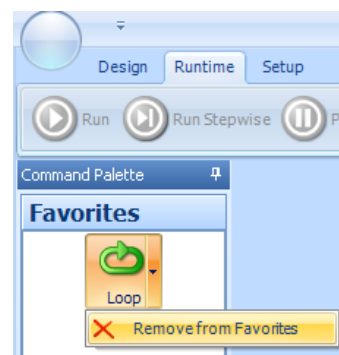
**Favorites empty**



**Loop, add to Favorites**



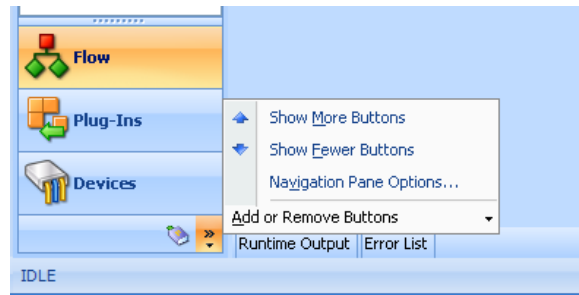
**Loop added to Favorites**



**Remove Loop from Favorites**

## 5.8 Palette configuration

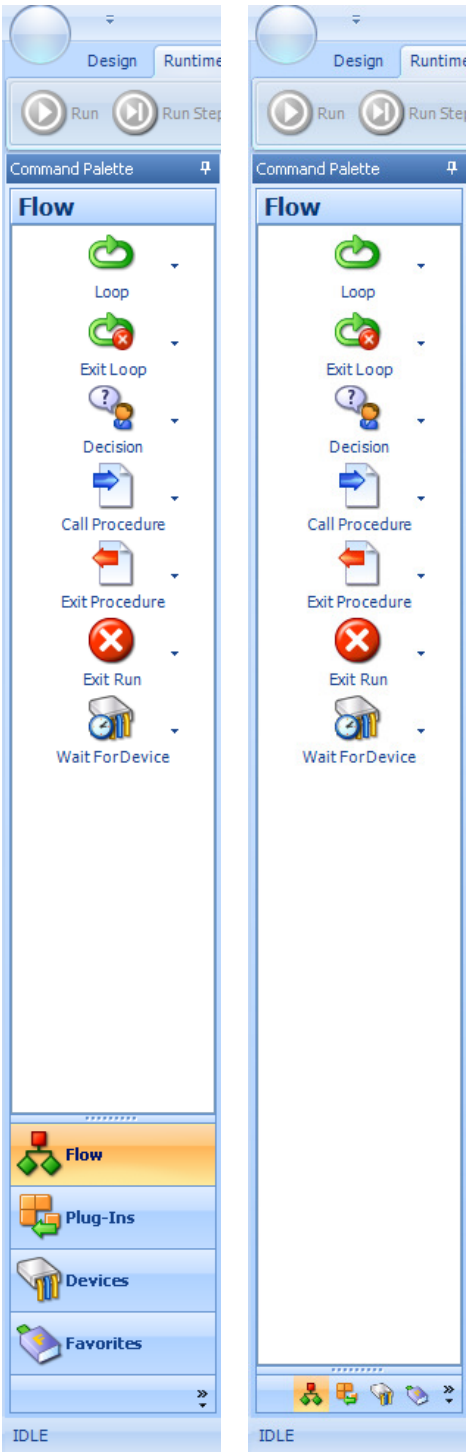
The palette layout can be configured.



**Palette options**

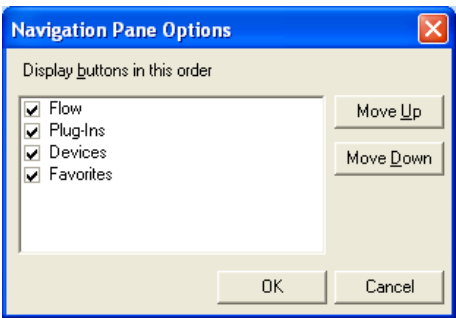
The options are:

- **Show More Buttons**, this will move the Favorites palette from an icon to a palette bar (Show More buttons)
- **Show Fewer buttons**, this will move palette bars to icons (Show Fewer Buttons)
- **Navigation Pane Options**, this allows ordering and removal of palettes as required (Navigation Pane Options)
- **Add or Remove Buttons**, easy addition and removal of palettes (Add or Remove Buttons)

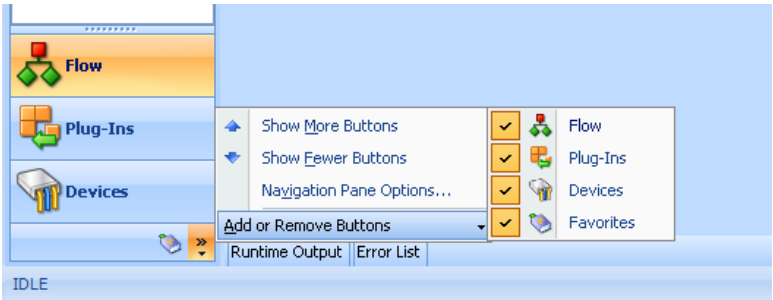


Show More buttons

Show Fewer Buttons



Navigation Pane Options



Add or Remove Buttons

## 5.9 Overlord3™ window controls

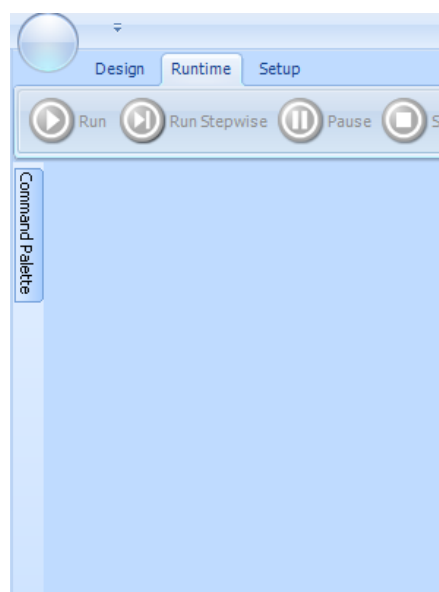
There are a number of icons available on the Overlord3 screen that allows manipulation of the environment and work area.

### 5.9.1 Environmental controls, thumbtacks

The thumbtack in the top right of a panel (Thumbtack selection) allows that information panel to be converted to an auto-hide tab. Pass the mouse over the thumbtack and click and the panel will disappear off the screen and be replaced with a tab (Replacement tab).



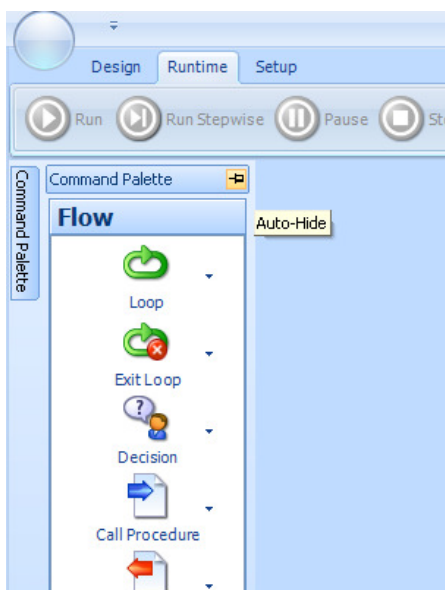
**Thumbtack selection**



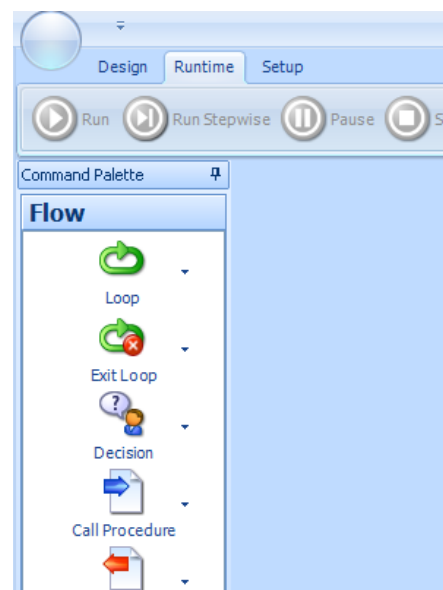
**Replacement tab**

The panel can be made visible, and used, by passing the mouse over the tab (select tab and thumbtack to make panel visible). The panel can be made visible permanently by passing the mouse over the tab and clicking on the thumbtack (panel visible again).



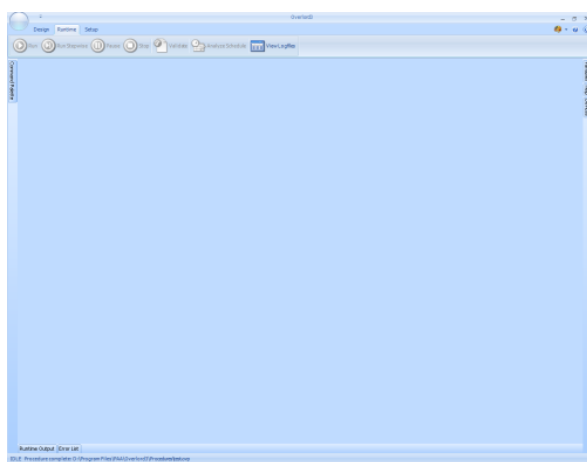


**select tab and thumbtack to make panel visible**

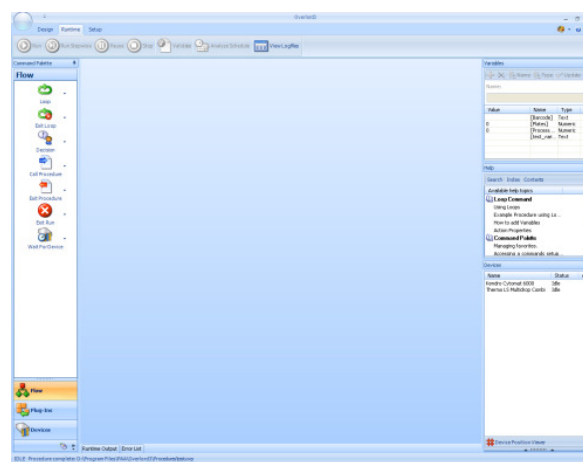


**panel visible again**

All the thumbtacks can be selected and the resulting tab makes a very clean interface (full thumbtack selection) compared to displayed panels (before thumbtack selection).



**full thumbtack selection**

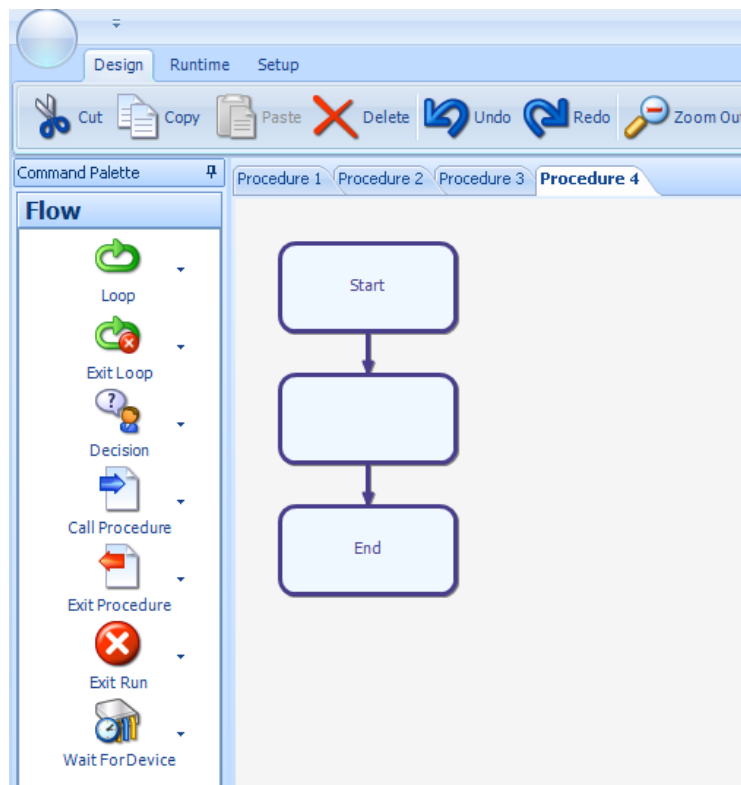


**before thumbtack selection**

## 5.9.2 Work Area controls, Multiple Document Interface

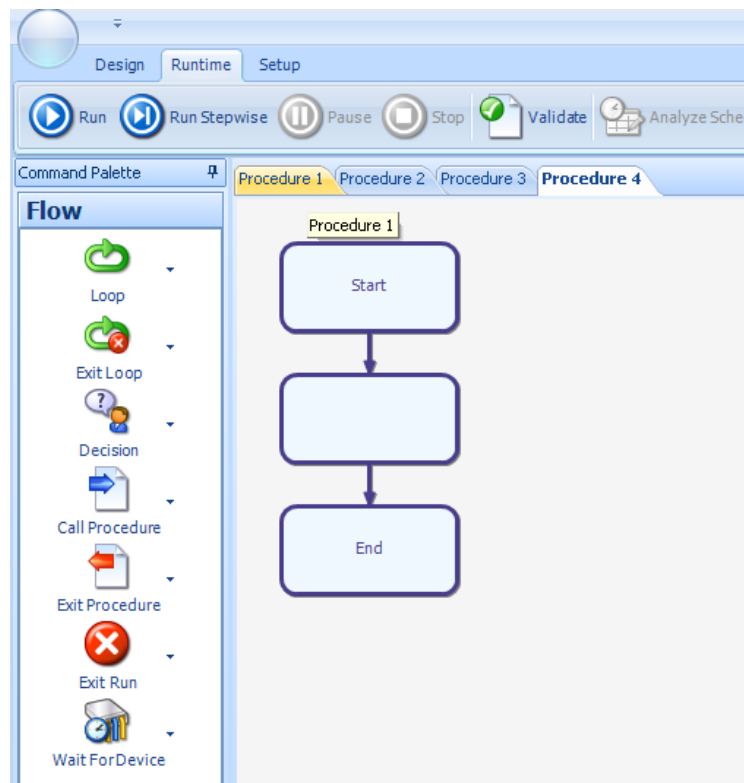
Overlord3 has a flexible multiple document interface. This means the user can have many Overlord3 procedures on the screen simultaneously, with the option of each procedure being displayed or on a tab.

For example, open four new procedures (Selection 5.2.1 File options) to get four procedures to manipulate (Load four procedures).



## Load four procedures

The presentation of the procedures to the user can be manipulated. To see **Procedure 1** and **Procedure 4**, click on the **Procedure 1** tab and keep the mouse button down (Grab the Procedure 1 tab).



**Grab the Procedure 1 tab**

Move the mouse and the location icon will appear (windows location icon). The user can choose the location where the window will reside (Figure 19 - add new window at top of screen, Figure 20 - add new window at right of screen, Figure 21 - add new window at bottom of screen, Figure 22 - add new window at left of screen, Figure 23 - add new window to tab list, Move mouse and location icon will appear).



**windows location icon**



**Figure 19 - add new window at top of screen**



**Figure 20 - add new window at right of screen**



**Figure 21 - add new window at bottom of screen**

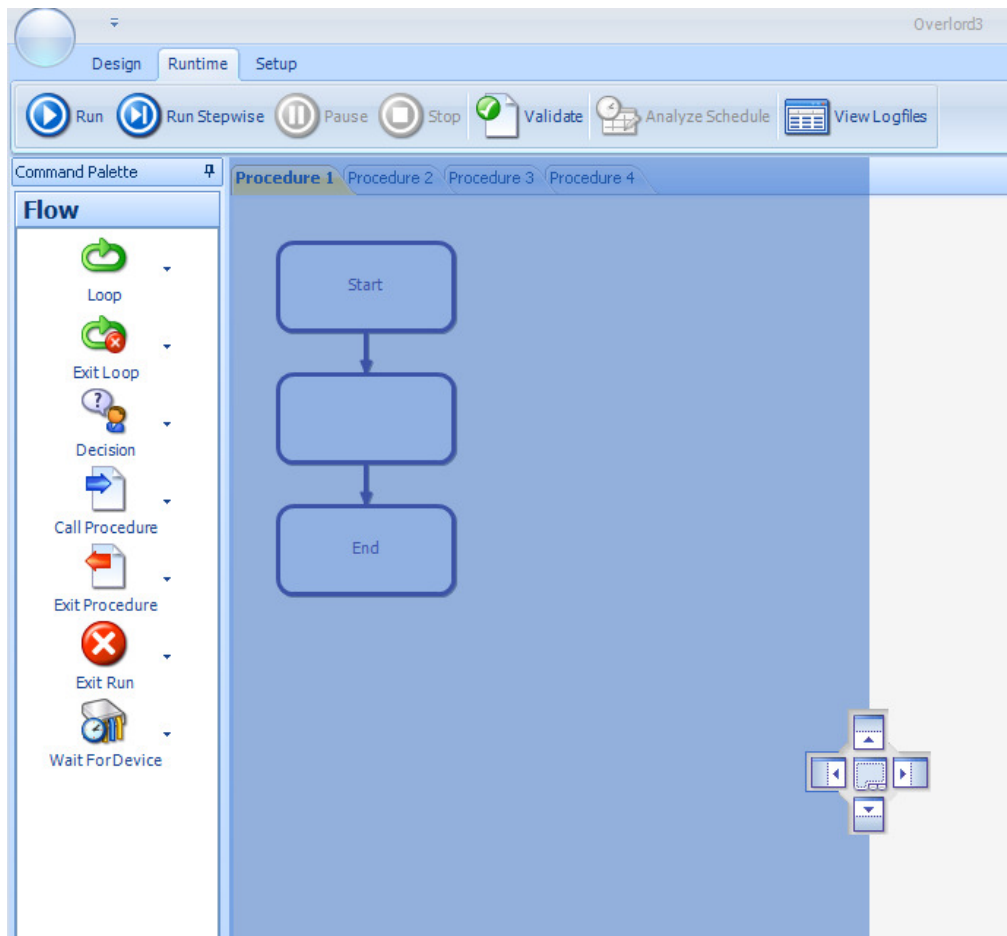


**Figure 22 - add new window at left of screen**



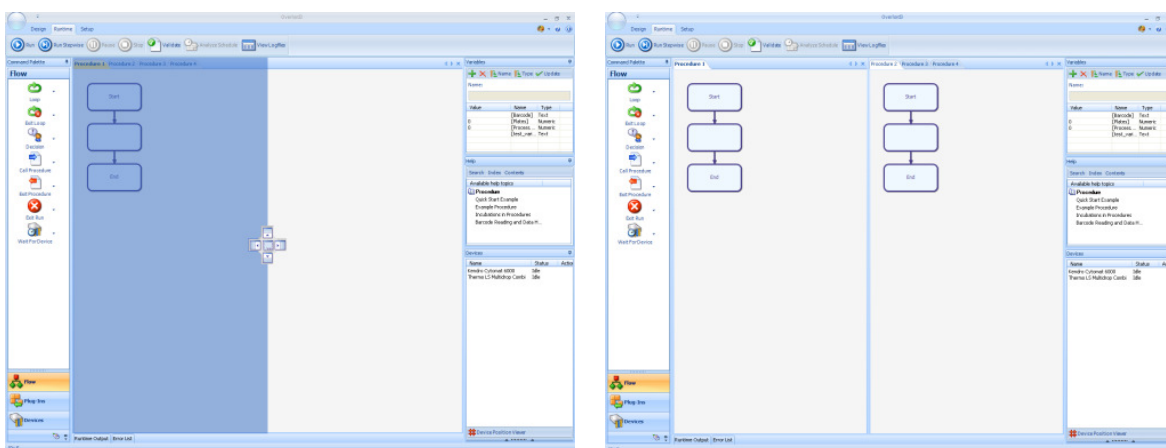
**Figure 23 - add new window to tab list**

When an icon is selected, the area is greyed out to show the user what the screen might look like if that option is selected.

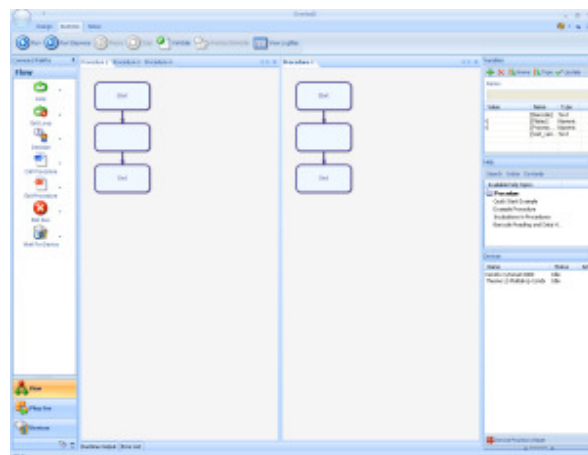
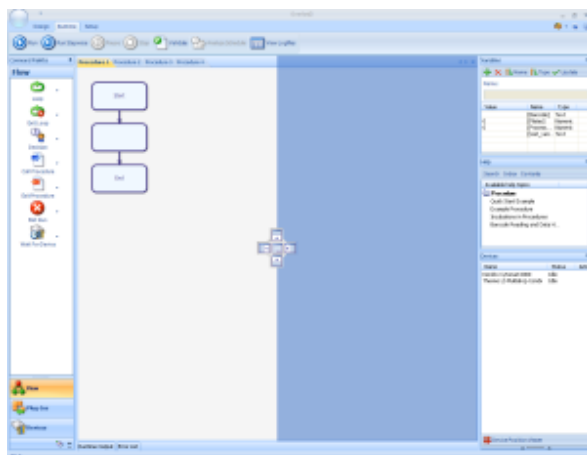


**Move mouse and location icon will appear**

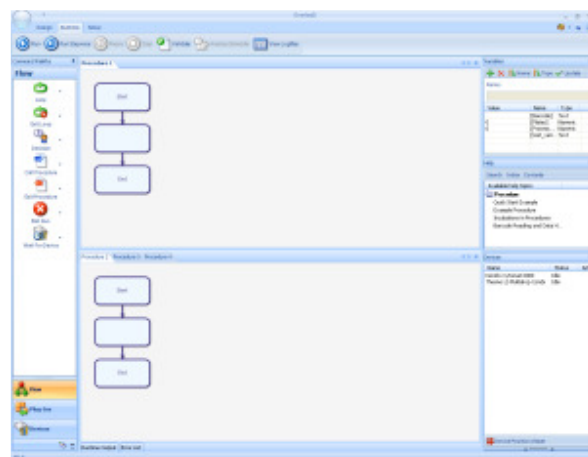
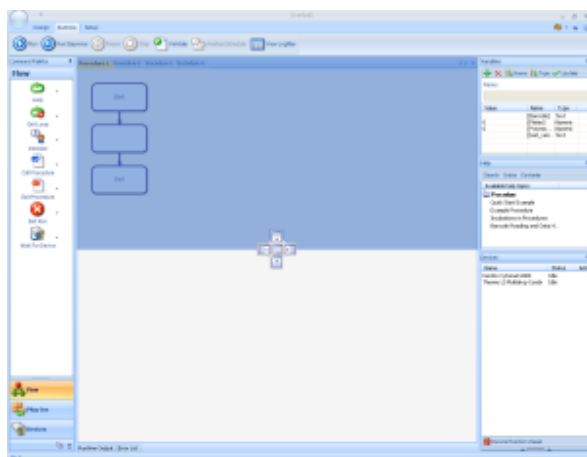
Examples of positioning left (position left), right (position right), up (position up) and down (position down) are shown below.



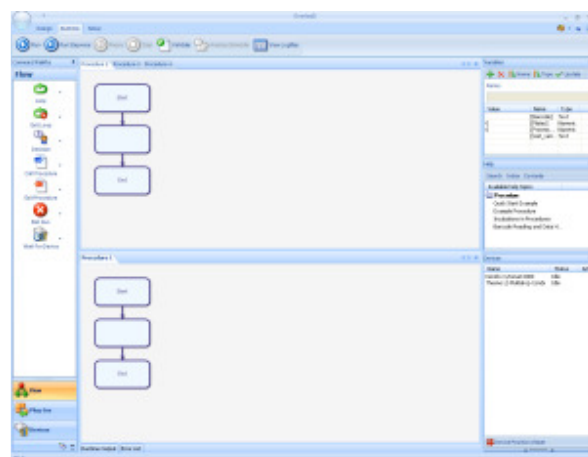
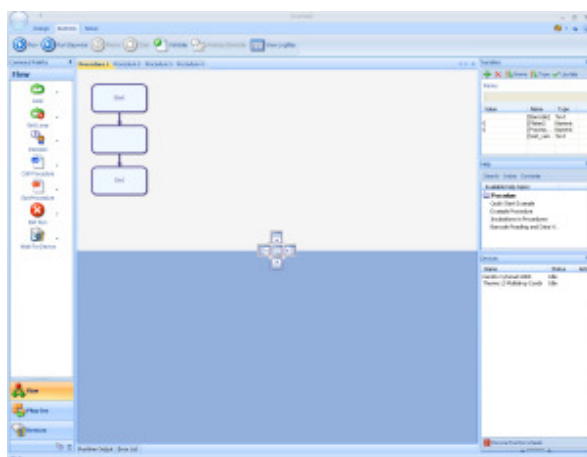
**position left**



**position right**

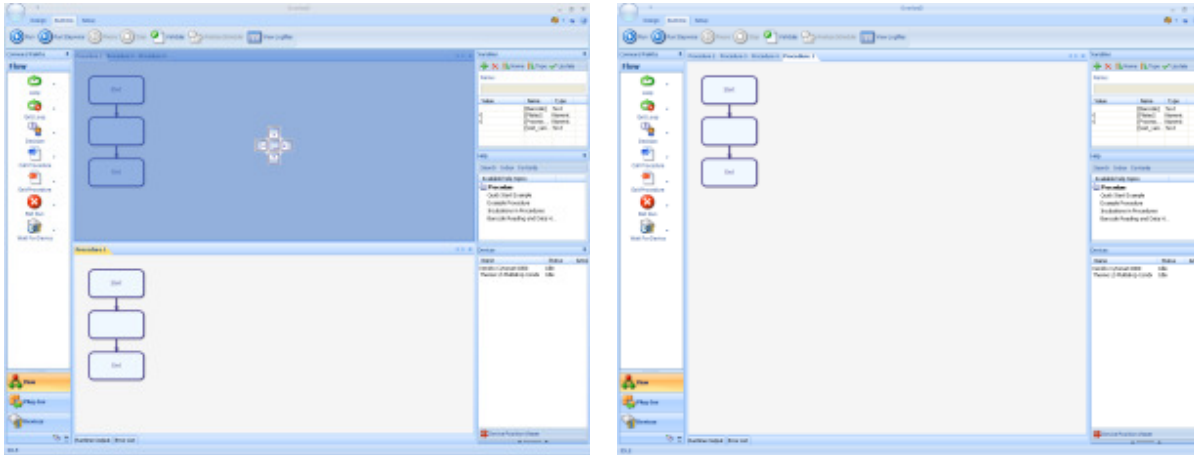


**position up**



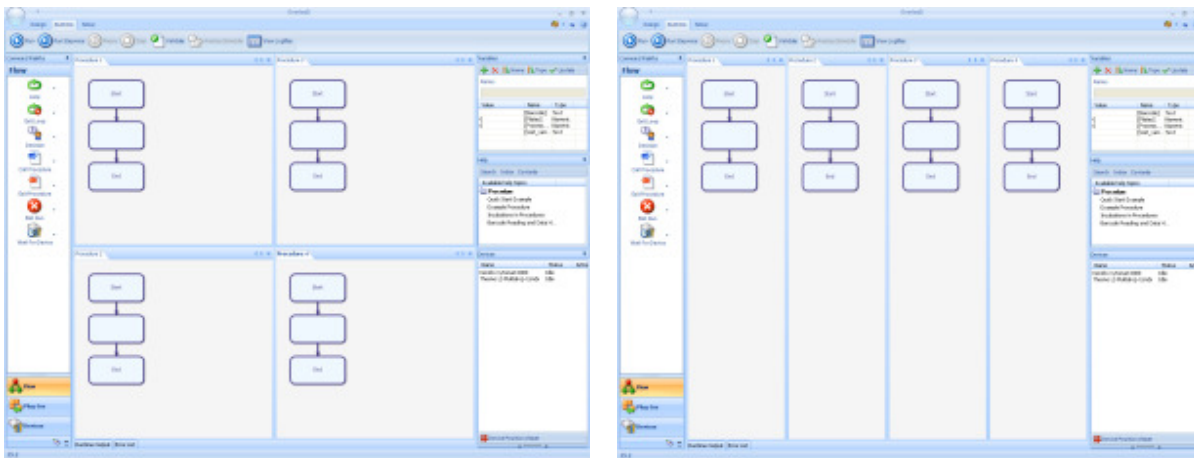
**position down**

The **Procedure 1** tab can be repositioned in the tab row by dragging the tab to the centre of the positioning icon (return to the tab row).



**return to the tab row**

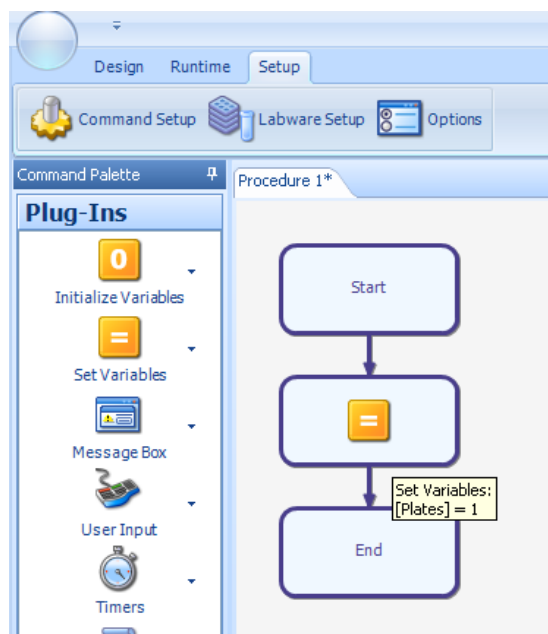
The use of these tools can position the procedures in a user favourable way. Examples are shown below (different page payouts for 4 procedures).



**different page payouts for 4 procedures**

## 5.9.3 Tooltips

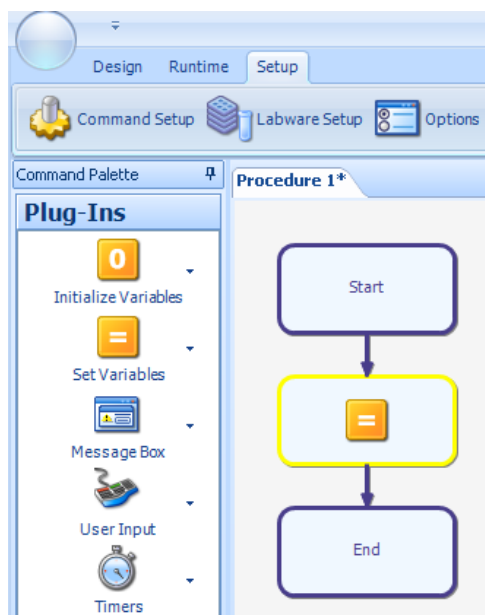
By passing the cursor over any of the flow chart steps, Overlord3 will show what that step will do at runtime by displaying a tooltip (Tooltips after passing mouse over procedure step).



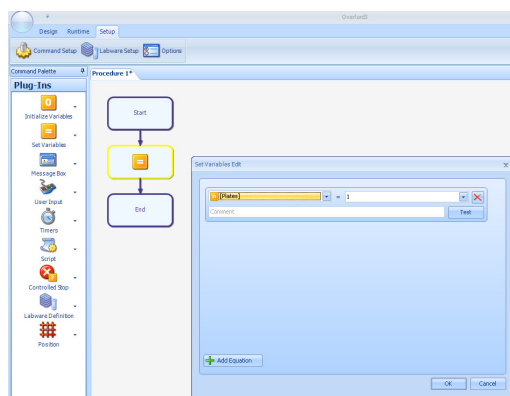
**Tooltips after passing mouse over procedure step**

## 5.9.4 Getting and changing properties of a procedure step

To change the parameters of a procedure step, the user should select the procedure step (Changing the properties of a procedure step, double click) and double click to display the properties (Dialog box to view/change properties).



**Changing the properties of a procedure step, double click**



**Dialog box to view/change properties**

The user can then view and or make changes to the properties of the procedure step before selecting:

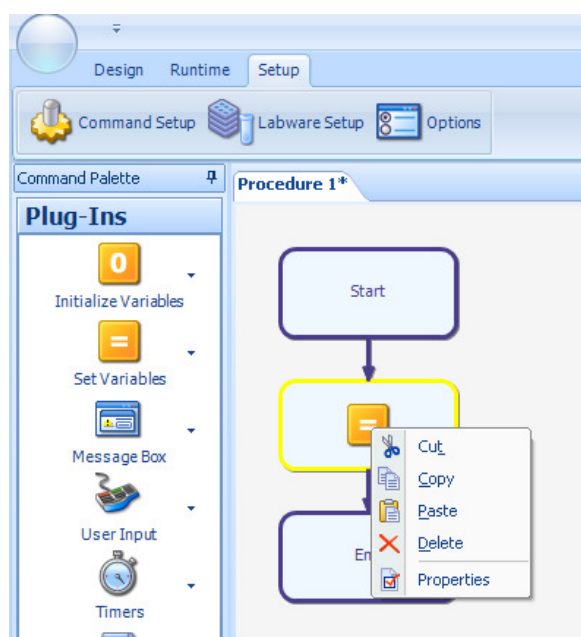
**OK** to exit and save the changes.

**Cancel** to ignore the changes.



## 5.9.5 Cut, Copy, Paste, Delete, Properties

In addition to the toolbar functions to edit the procedure flow chart, the user can also use the mouse edit function by passing the mouse over the required step and single right hand mouse click hand mouse click (Mouse editing menu).



**Mouse editing menu**

The options are described in other sections of the manual:

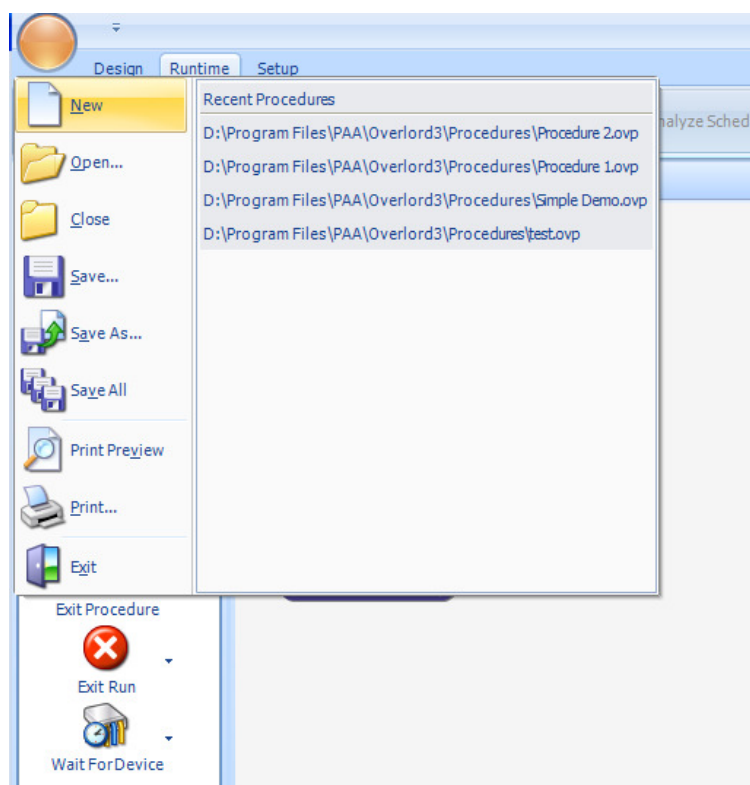
- Cut (see Cut)
- Copy (see Copy)
- Paste (see Paste)
- Delete (see Delete)
- Properties (see Properties)

## 6 Creating Procedures

**Procedures** are created by dragging **Commands** across onto the flowchart to build up **Actions**. The way the procedure runs Actions can then be controlled by using Overlord variables. The following sections describes how all of these components work together to run an Assay and is followed with an example on how a simple procedure can be built up.

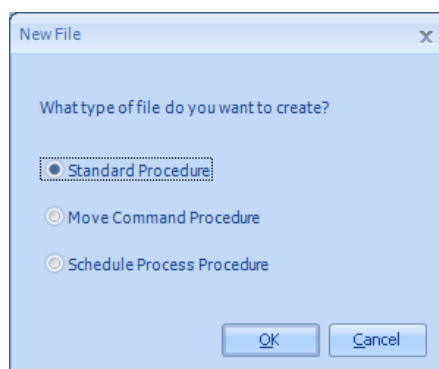
### 6.1 Opening a new procedure

From the circle menu (**Overlord3™** screen, top left) select **New** (New Procedure).



**open a new procedure**

The user is offered a choice of procedure type to create (Choose procedure type).



## Choose procedure type

There are three types of procedure:

- Standard Procedure
- Move Command Procedure
- Schedule Process Procedure

See **scheduling section** for details on the different scheduling types.

## 6.2 Standard Procedure

A standard procedure produces an **event driven** procedure. These procedures have the complete set of Overlord3 functions available.

These procedures are identified with the **<filename>.ovp** filename type and are saved in the normal way.

## 6.3 Move Command Procedure

These are automated moves that are used by the **Schedule Procedure** type process. The **Schedule Procedure** type reduces the user input by inferring that a process that has a diluter as stage 1 and a plate reader as stage 2 will require a robot or stacker move to transfer the plate from the diluter to the plate reader. These moves are called **Move Commands** and defined in the **Move Command Procedure** function.

Not all **Overlord3™** commands are available in a **Move Commands Procedure**, please see the list below.

<b>Flow</b>	<b>Loop</b>	✓
	<b>Exit Loop</b>	✓
	<b>Decision</b>	✓
	<b>Call Procedure</b>	✓
	<b>Exit Procedure</b>	✓
	<b>Exit Run</b>	✓
	<b>Wait For Device</b>	✓
<b>Plug-Ins</b>	<b>Initialise Variable</b>	✗
	<b>Set Variables</b>	✓
	<b>Message Box</b>	✗
	<b>User Input</b>	✗
	<b>Timers</b>	✓
	<b>Script</b>	✓
	<b>Controlled Stop</b>	✓
	<b>Labware Definition</b>	✓
	<b>Position</b>	✓
<b>Devices</b>	<b>All</b>	✓

This allows **Move Commands Procedure** complex functions to be performed, for example, if there is a lid on the plate remove and park the lid, before placing the plate on an instrument.

**Move Commands Procedures** don't have a runtime mode.

These procedures are identified with the **<filename>.mcp** filename type and are saved in the normal way.

## 6.4 Schedule Process Procedure

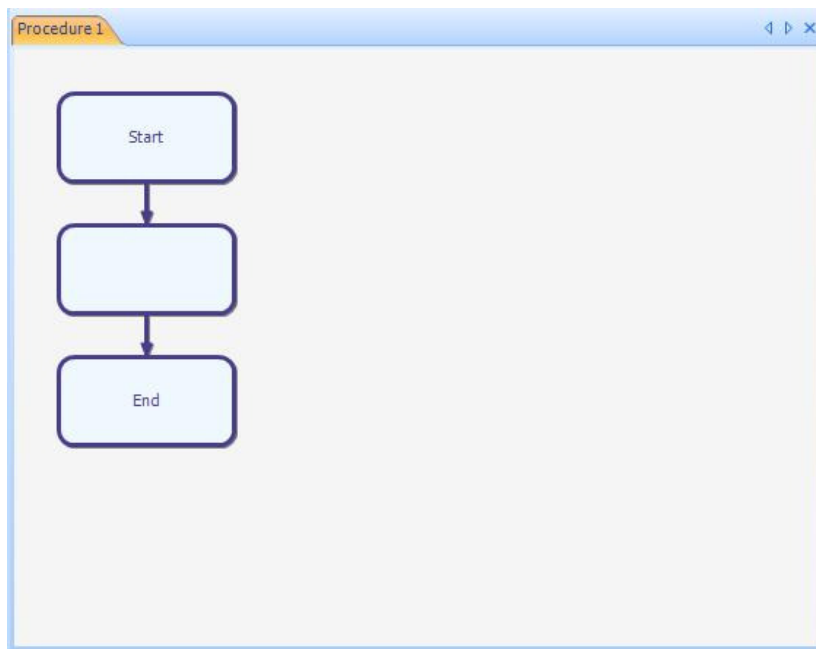
A **Schedule Process Procedure** is a time driven scheduler that will run a dynamic scheduling mode.

The time dependency of the process of the standard **Overlord3** means that **Overlord3** functions that might lead to a non-determinant time frame are not available. Non-determinant times make the initial scheduling process impossible.

<b>Flow</b>	<b>Loop</b>	✓
	<b>Exit Loop</b>	✗
	<b>Decision</b>	✓
	<b>Call Procedure</b>	✓
	<b>Exit Procedure</b>	✗
	<b>Exit Run</b>	✗
	<b>Wait For Device</b>	✗
	<b>Suspend Process</b>	✓
<b>Plug-Ins</b>	<b>Initialise Variable</b>	✗
	<b>Set Variables</b>	✓
	<b>Message Box</b>	✗
	<b>User Input</b>	✗
	<b>Timers</b>	✗
	<b>Script</b>	✓
	<b>Controlled Stop</b>	✓
	<b>Labware Definition</b>	✗
	<b>Position</b>	✗
<b>Devices</b>	<b>All</b>	✓

## 6.5 Procedure Flowchart

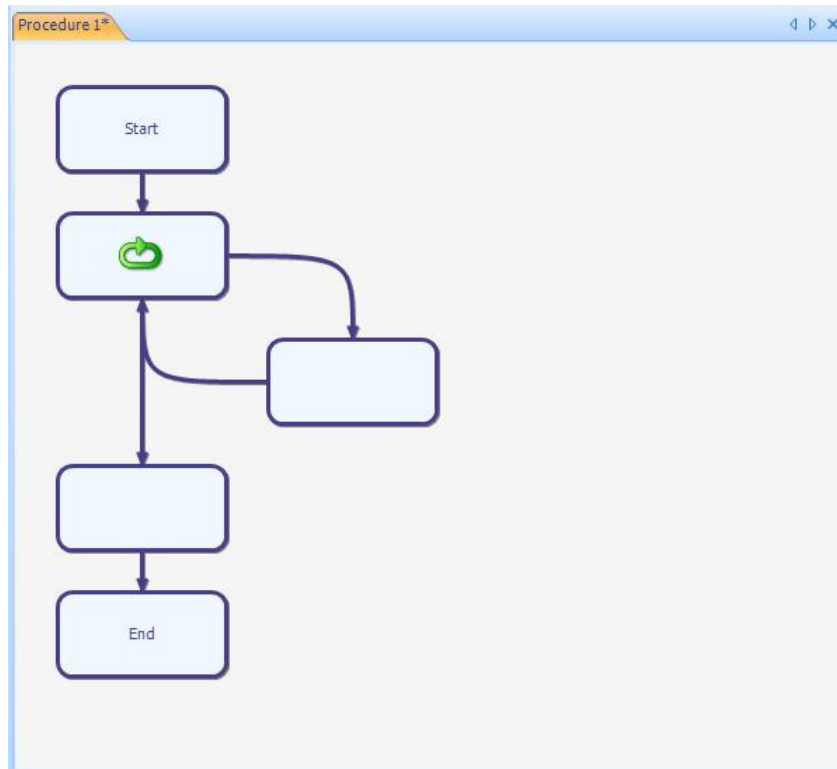
When Overlord3 starts, it opens with an empty procedure that consists of a **Start** Action, an empty Action and a **Finish** Action. All procedures contain a Start and Finish Action and these cannot be removed. Commands are dragged in between these two Actions to build up the procedure. A new procedure, if there isn't already one open can be created by going to the File Menu and selecting 'New Procedure'. This should load the following window (New Procedure).



**New Procedure**

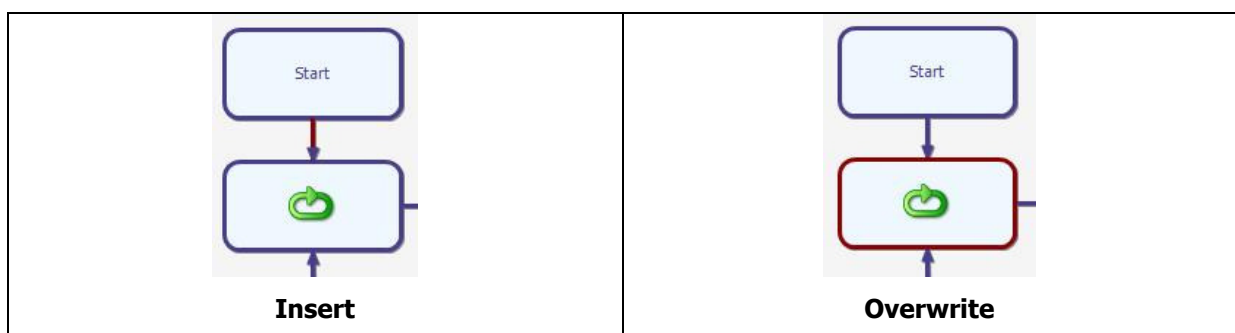
A Command can then be added into the empty Action box or inserted between existing Action boxes. The place where the command will be added will be highlighted so when the mouse button is released, it is clear where the command and resulting Action will be placed.

If a Loop command is inserted between the Start action and the Empty Action, the procedure will expand to include this new Action (Add Loop Command).



**Add Loop Command**

Add a Message Box command onto an empty box and the procedure remains the same, the only difference being the Action in that place is updated. If you add a command onto a box that already contains an Action, a dialog will appear asking if you want to overwrite the existing Action. The basic rules are, dragging onto an arrow will insert the Action (Insert), dragging onto a box will overwrite the Action (Overwrite).



Existing Action boxes can also be moved within a procedure or deleted.

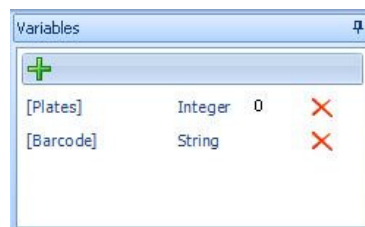
## 6.6 Overlord Variables

Overlord variables have a very important role in Overlord3. They control how procedures execute at runtime and they pass data around between devices on the system and with the outside world.

There are 2 default variables on every system, [Plates] and [Barcode]. They have the following uses.

Variable	Use
<b>[Plates]</b>	Represents how many plates are to be run. Typically used to control how many times a Loop action should loop around.
<b>[Barcode]</b>	Normally used within a Barcode action. The read barcode is stored in this variable. This can then be used in a script to rename a file with the barcode of the plate.

The current list of variables on the system is listed in the Variables pane which is normally in the top right of the main window (Variables pane).



**Variables pane**

Extra variables can be added by clicking the green cross. The name of the new variable and its type can then be added (Add Variable).



**Add Variable**

The type of an Overlord variable refers to the type of data it contains. There are four types of variable available.



Type	Description
<b>Numeric</b>	For variables that store numbers such as the current plate or the volume to dispense.
<b>Text</b>	For variables that store text such as the data path to write new generated plate data to.
<b>True/False</b>	Variable that is either true or false. Used typically for variables that indicate if the plates have lids or not.
<b>Object</b>	This variable type is used on more advanced setups and is used in association with Scripts. It allows Objects that were created in a Script to be persisted across multiple scripts. An example would be an instance of the Excel Application Object. This would mean this object would only need to be created once if we were using multiple times within a procedure.

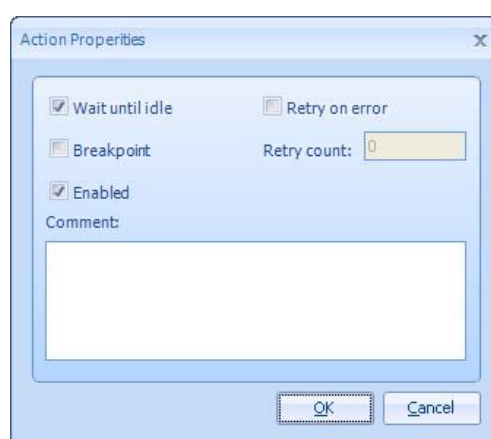
## 6.7 Actions

Each time a Command is dragged onto the Procedure flowchart, an Action is created. At runtime, each Action will execute a specific task using the Command it was created with.

If a Robot command (e.g. Mitsubishi Robot) is dragged onto the flowchart an Edit screen will appear for that Command. If the robot program "Get Plate From Incubator" is selected, then at runtime, that program will be executed.

Alternatively, if a Loop command is dragged onto the flowchart and Edit screen will appear for setting options on how to loop. If the loop is set to loop around 3 times, at runtime, it will do exactly that. Refer to Section 6.7 Actions for details on how all the Overlord3 core commands work. Refer to the Readme files that come with each Overlord Driver, available from the PAA website , for details on what options are available.

Each Action is similar in that they each have a set of properties. By highlighting an action and then right mouse clicking, the context menu will appear. At the bottom of this menu will be the entry "Properties". Selecting this will bring up the Action Properties dialog (Action Properties).



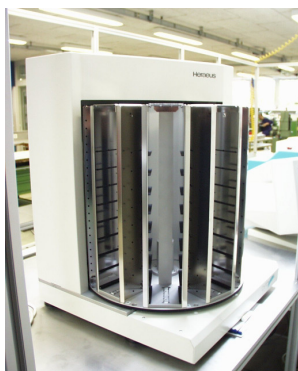
**Action Properties**

The table below describes each of the properties that are available and what they do.

Property	Description
<b>Wait until Idle</b>	By default, when an Action is executed at runtime, the procedure execution will wait for the Action to complete. If this property is turned off, the Action will be started, but the procedure will carry on running moving onto the next Action. If the procedures are built up properly, Actions representing Devices (instruments) can be run in parallel which improves the throughput of the system that Overlord3 is running.
<b>Breakpoint</b>	If the Breakpoint property is set, the execution of the procedure will pause when it comes up this Action. This feature is typically used when testing the procedure for the first time. Typically, you turn a breakpoint on, on a Robot Action that was about to move a plate into an area of the Workcell where a collision could occur if the devices was not in the right state (e.g. a pipetting station with its arm in the position where the plate would be loaded)
<b>Enabled</b>	By default, an Action is enabled. If this property is turned off, at runtime it will not be executed.
<b>Retry on error</b>	If an error is generated on this Action at runtime and this property is turned on, the Action will automatically run again until an error is actually generated. The number of times it 'Retries' is set in the text box labelled 'Retry Count'
<b>Comment</b>	The comment box allows a better explanation of what the Action is doing, to be associated with it.

## 6.8 Example Procedure

This section will go through step-by-step how to create a procedure that will move a set of plates from a Thermo Kendro Carousel (Thermo Kendro Cytomat Carousel) onto a Molecular Devices Aquamax (Molecular Devices AquaMax) using a Mitsubishi RV-2AJ (Mitsubishi RV-2AJ robot).



**Thermo Kendro Cytomat Carousel**



**Molecular Devices AquaMax**



**Mitsubishi RV-2AJ robot**

The procedure will be setup as follows.

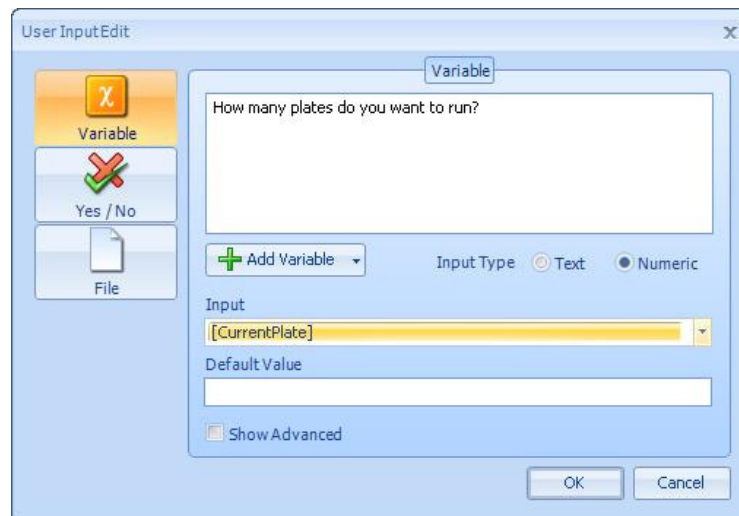
1. There will be a main procedure that uses sub-procedures that move plates to and from the Aquamax
2. The procedure will allow a variable number of plates to be run
3. The procedure will allow the user to enter which program to run on the Aquamax
4. The Kendro will be setup to automatically retry on error
5. Each robot action will have break points for testing purposes

It is assumed that Overlord3 is already setup with devices described above. The commands (drivers) for these devices can either be downloaded from the PAA website or by contacting PAA directly. It is also assumed that the Mitsubishi has already been programmed with the following 4 programs

- GPCYT – Get Plate From Kendro Cytomat
- PPCYT – Put Plate onto Kendro Cytomat
- GPAQ – Get Plate from MD Aquamax
- 1. PPAQ – Put Plate onto MD Aquamax

The procedure requires 2 new **Numeric** variables. Add a variable called '**CurPlate**' and another called '**AquamaxProgram**'. Refer to Section 6.6 on how to add new variables.

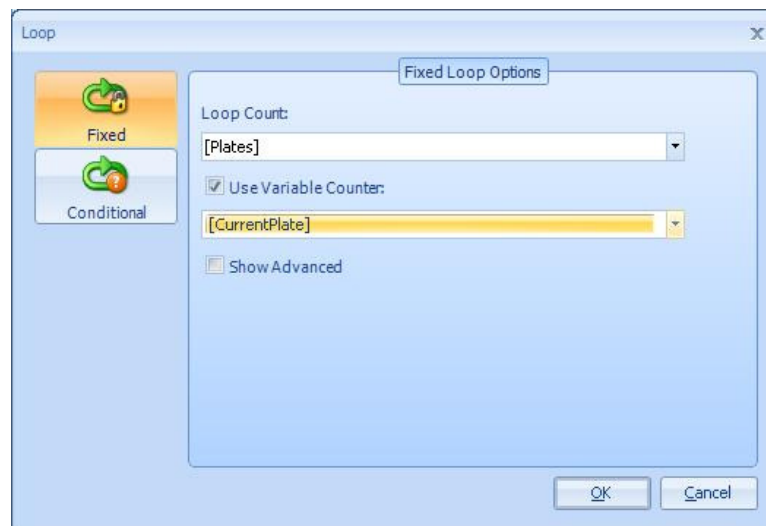
2. Starting with a new Procedure, add a **User Input** command that asks the user how many plates they want to use. Store this value into **[Plates]** (User Input for Number of Plates).



### User Input for Number of Plates

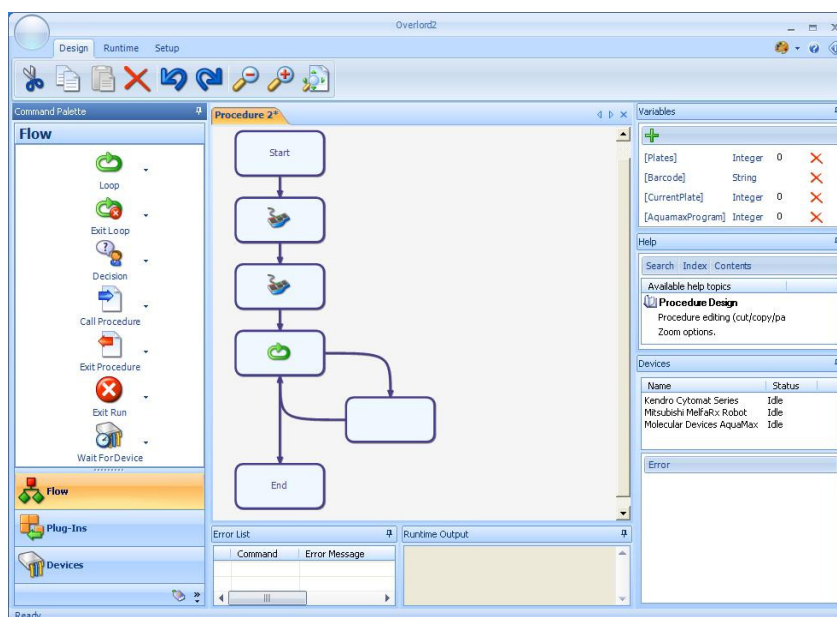
3. Follow this with another **User Input** command that asks the user what Aquamax Program they want to run. Store this value into **[AquaMaxProgram]**.

Add a **Loop** command after the last User Input action and configure this to be a fixed loop that loops **[Plates]** times. Set it up so that the variable **[CurrentPlate]** is updated with the current Loop number (Fixed Loop).



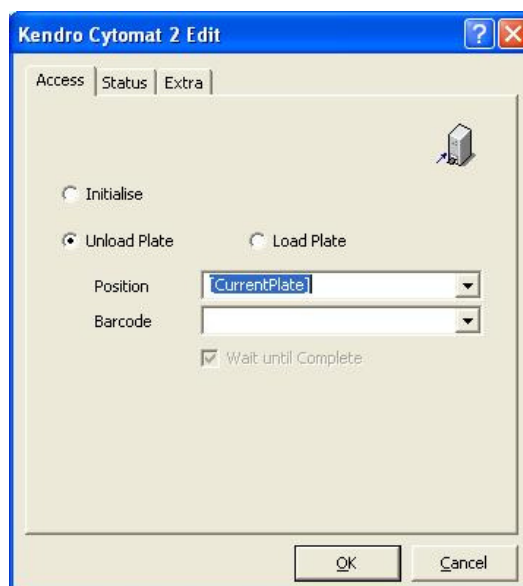
### Fixed Loop

4. At this point the procedure should contain 2 User Input commands and an empty Loop (Empty Loop). The next step is to fill in the loop with the Actions that will be run on each plate



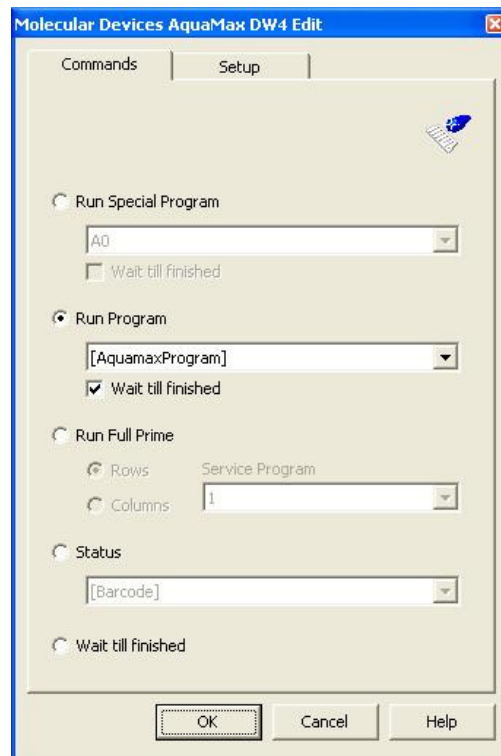
Empty Loop

5. Add a **Kendro Cytomat Series** command from the Devices palette into the loop (Setup Cytomat to unload). Select the option to Unload a plate from the position **[CurrentPlate]** (Setup Cytomat to unload). If the loop is setup to loop 5 times, then the first time through the value of [CurrentPlate] will equal 1. 2<sup>nd</sup> time through it will equal 2, 3<sup>rd</sup> time through it will equal 3 and so on until the final loop through, the 5<sup>th</sup> time through where [CurrentPlate] will equal 5. With this setup, each time we go through the loop, a different plate is unloaded from the Kendro Cytomat.



Setup Cytomat to unload

6. Next add a **Mitsubishi MelfaRX** command to pick up the plate from the Kendro since at this point, the plate would be sitting on the Kendro Transfer station. Select the robot program to pickup a plate which in this case is **GPCYT**.
7. Add another **Mitsubishi MelfaRX** command to place the plate onto the Aquamax. Select the program **PPAQ**.
8. Add a **Molecular Devices Aquamax** command to run a program on the Aquamax (Run Aquamax Program). This should be configured so that it runs a program represented by the variable **[AquamaxProgram]**. When selecting the program make sure the 'Wait until Finished' checkbox is selected. It is very important this checkbox is selected when using Overlord1 Drivers in Overlord3.

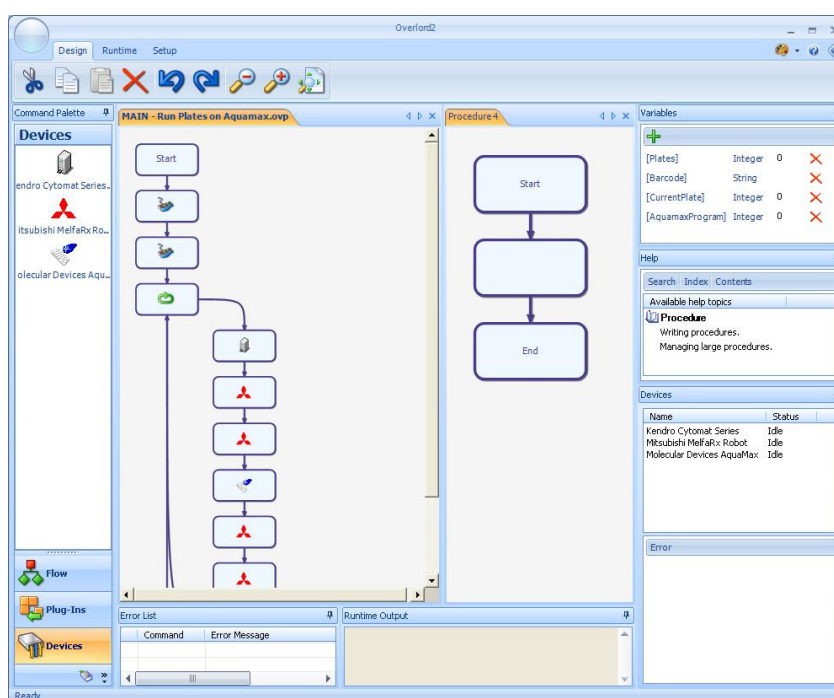


**Run Aquamax Program**

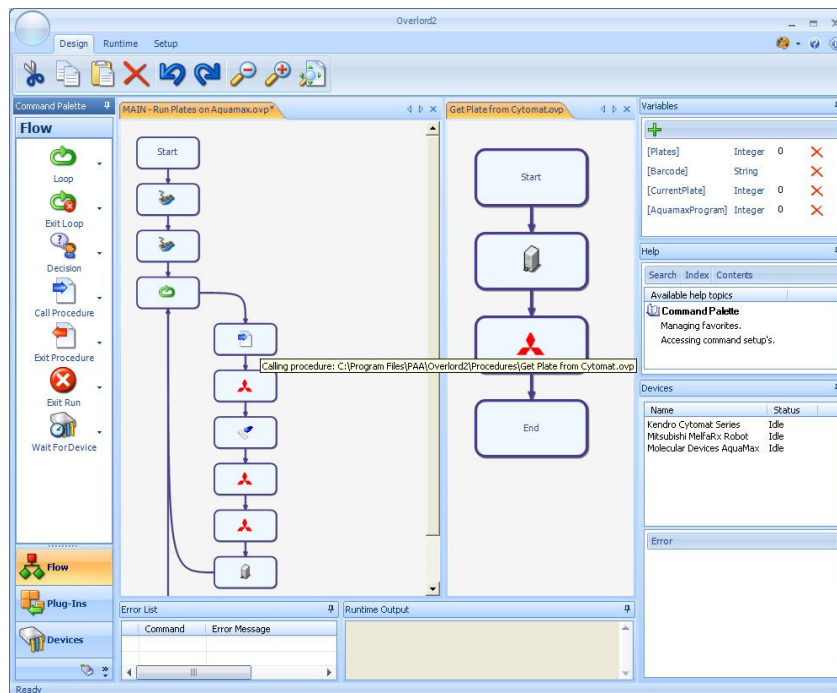
9. At this point in the run, the Aquamax would have finished running the program so the plate can be returned to the Cytomat. 3 commands are required for this. Two **Mitsubishi** commands are required to pickup the plate, **GPAQ**, and place onto the Kendro, **PPCYT**. Finally the plate can be returned into the position within the Kendro it originally came from. Add a Kendro Cytomat Series command to load the plate into position **[CurrentPlate]**
10. The program is now complete and should be saved. Save this as MAIN - Run Plates on Aquamax.ovp. The Kendro Cytomat actions need to be setup to retry automatically upon error plus a breakpoint is needed on the first Mitsubishi action. Select the first Kendro Cytomat set the Retry on error property so that if there is an error the Cytomat will automatically retry before reporting an error (See section 6.7 Actions). Repeat for the other Cytomat action. Select the first Mitsubishi command and set the 'Breakpoint' property.

11. At this point, all of the Actions are with one procedure. One of the requirements for building up this procedure was to use sub-procedures for moving plates to and from the Cytomat. Imagine if we also on this system had another procedure that was identical to the procedure created above, but instead of an Aquamax it used a Biotek Washer. If down the line we wanted to add in a barcode reader step, we would need to modify both procedures to add this change. However, if we use sub-procedures, we only need to make this change in the sub-routine since both Main procedures would be using this. Building up sub-routines is good practice since it reduces the chance of writing the same blocks of actions over and over again.

Create a new procedure and dock this next to the existing procedure (Create Sub Procedures). Select the Cytomat Action and the Mitsubishi command that unloads a plate from the Cytomat and picks it up (Get Plate from Cytomat Sub-Procedure). Cut and paste these into the new procedure and save this as 'Get Plate from Cytomat'. Add a **Call Procedure** command into the Main procedure to call this new procedure.



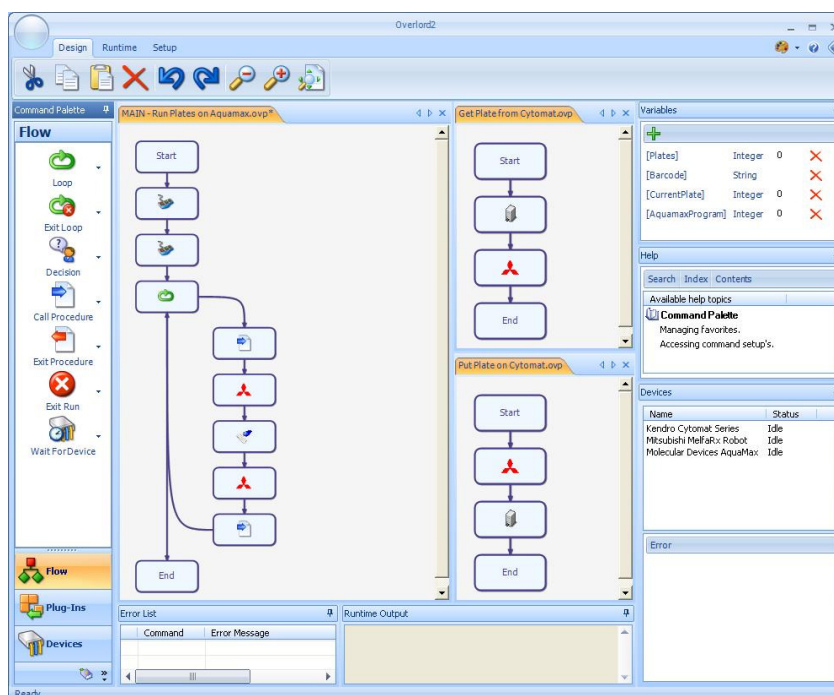
**Create Sub Procedures**



**Get Plate from Cytomat Sub-Procedure**

12. Create another new procedure and then remove the last Mitsubishi action and Cytomat action from the Loop in the main procedure. Cut and paste these into the new procedure and save this as 'Put Plate on Cytomat.ovp'. Add a **Call Procedure** command into the Main procedure where these two actions were removed that calls this new procedure.
13. The procedure is now complete, we have one main procedure, using two sub procedures, that moves a variable number of plates onto an Aquamax and runs an Aquamax program that is set by the user at runtime (The Final procedure(s)).





**The Final procedure(s)**

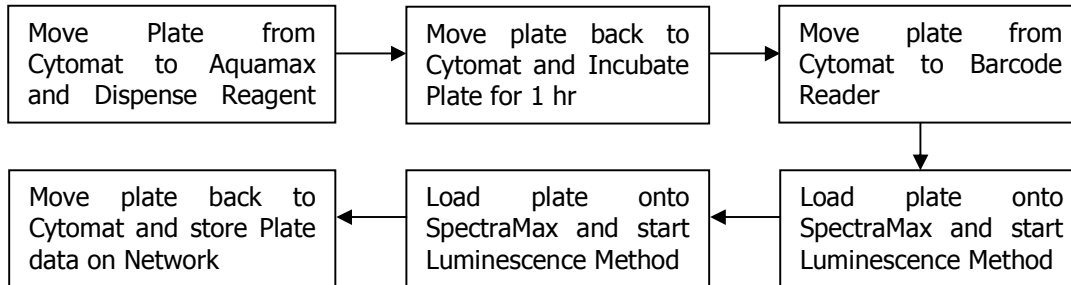
## ***6.9 Advanced Procedure with Incubations and Data Handling***

Incubations can be run in Overlord3 by using the Timers command. By allocating a Timer to a plate and telling it to run for a period of time, we are effectively doing the same as if the plate was being run manually using a stopwatch for the incubations. With a stopwatch, you wait until the required time has elapsed before carrying on with the next steps in the assay. In Overlord3, you do exactly the same, you tell the Procedure to wait until the timer in question has elapsed before moving onto the next set of Actions.

There are also other ways to monitor timers. Decisions and Loops can work against the state of a Timer. However, for this example we will just be using the Timer command option "Wait until Complete".

Taking the previous example (Section 6.8),

Example Procedure: this will be extended so that after the plate is run through the Aquamax it will be incubated for 1hr in the Cytomat before running it through a Molecular Devices SpectraMax using Softmax Pro. Finally, the plate will be moved back into the Cytomat. The plates will also now be bar-coded so a barcode reader is required. The data generated by the SpectaMax will need renaming with the barcode and stored onto a network resource for a LIMS to pickup. See the flow diagram below for an illustration of the assay we are looking to automate.



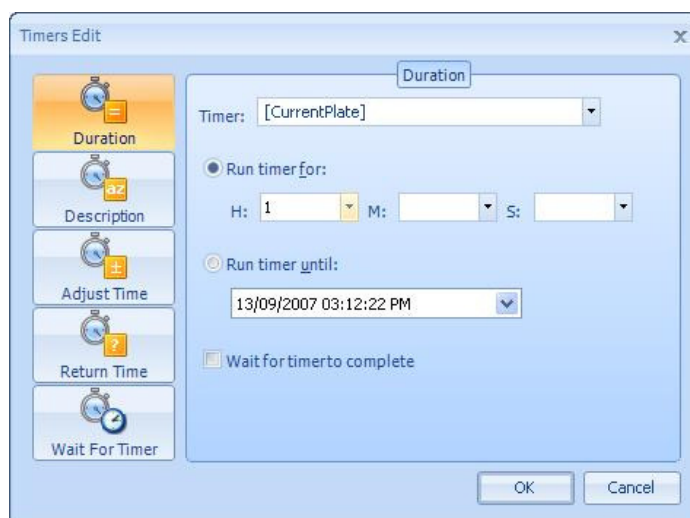
**Metrologic  
Barcode Reader**



**Molecular Devices SpectraMax Plus384  
Plate Reader**

Before modifying the existing procedure, the Metrologic Barcode reader and MD SpectraMax commands (See above) need to be downloaded from the PAA website and added. It is also assumed that the Mitsubishi has been programmed to additionally contain the following 3 programs.

- GPSM – Get Plate From SpectraMax
  - PPSM – Put Plate onto SpectraMax
  - BARCODE – Move Plate past barcode reader (always gripping the plate)
1. Open the procedure MAIN - Run Plates on Aquamax.ovp and save as **MAIN – Incubation Demo.ovp**
  2. Add a timer command after the step that puts a plate back into the Cytomat (Add Incubation Timer). This resulting Action will be telling the procedure that this plate needs to be incubated for one hour. Set this action so that it uses the Timer represented by **[CurrentPlate]** and sets it to run for 1hr. Using **[CurrentPlate]** means that on the first time around the loop when we are using Plate1, it will be assigned Timer 1. Plate 2 will have Timer 2; Plate 3 will have Timer 3 and so on.

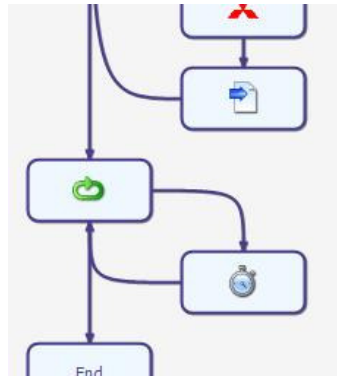


### Add Incubation Timer

- Once all the plates are back in the Cytomat they will each have their own timer, each one elapsing slightly later than the next. If we were running 5 plates, 5 timers would have been created. Assuming the first plate was loaded into the Cytomat at 10.00am and the time to move a plate to the Aquamax, run a program and back to the Cytomat takes 2 minutes, the Incubation elapse cycle would look as follows.

Plate Timer	Elaste Time (Incubation time of 1hr)
<b>Plate Timer 1</b>	<b>11:00 am</b>
<b>Plate Timer 2</b>	<b>11:02 am</b>
<b>Plate Timer 3</b>	<b>11:04 am</b>
<b>Plate Timer 4</b>	<b>11:06 am</b>
<b>Plate Timer 5</b>	<b>11:08 am</b>

The next part of the Overlord procedure needs to wait for relevant timer to elapse before moving the plate to the SpectraMax. Add another loop into the procedure after the existing loop that again loops **[Plates]** times and uses **[CurrentPlate]** to represent the current loop number (Add new Loop). Add a Timer command inside the loop and select 'Wait for Timer'. This should wait until the timer represented by **[CurrentPlate]** has completed. The desired effect here will be that the procedure will not move plates to the Plate reader until the Incubation has completed.



**Add new Loop**

4. The steps to move the plate to the reader and back need to be added after the Timer command. Add a Call Procedure command and select the OVP file '**Get Plate from Cytomat.ovp**' which was created in the previous example'.
5. Add a Mitsubishi command to read the plate on the barcode. Select the Robot program BARCODE. Follow this with a Metrologic Orbit command to take the barcode that was just read and store it into the Overlord Text variable [**Barcode**].
6. After this add a Mitsubishi command and select the program PPSM. It is assumed that the SpectraMax drawer is already open at this point. Follow this with the SpectraMax command. Select a method that has already been created in Softmax Pro, the software for running the reader in stand-alone mode. Set it up so that with each read the data file **C:\PlateData\Data.txt** is created.
7. Once the reader has finished, the drawer will open automatically. Add a Mitsubishi command to pick up the plate, PPSM. Add another Call Procedure command after this and select the program 'Put Plate on Cytomat.ovp'

All of the instrument calls are now complete. The only thing is missing is a way of renaming the generated plate data with the barcode of the plate and then moving this onto a network resource. This can be done by using a Script command. Add this after running the SpectraMax. Assuming the command has been setup to use VBScript use the following code.

```

Set objFSO = CreateObject("Scripting.FileSystemObject")
objFSO.CopyFile "C:\PlateData\Data.txt", "\\LIMS\AssayData\" + _
[Barcode] + ".txt", True

```

8. The procedure is now complete. This example shows how incubations can be added into an Overlord procedure by using timers. The entire assay has effectively been divided into 2 sub processes, **Run plates through Aquamax and Incubate** followed by **Run plates through Spectramax**. A general rule of thumb is that the number of incubations plus 1 is the number of sub processes required.

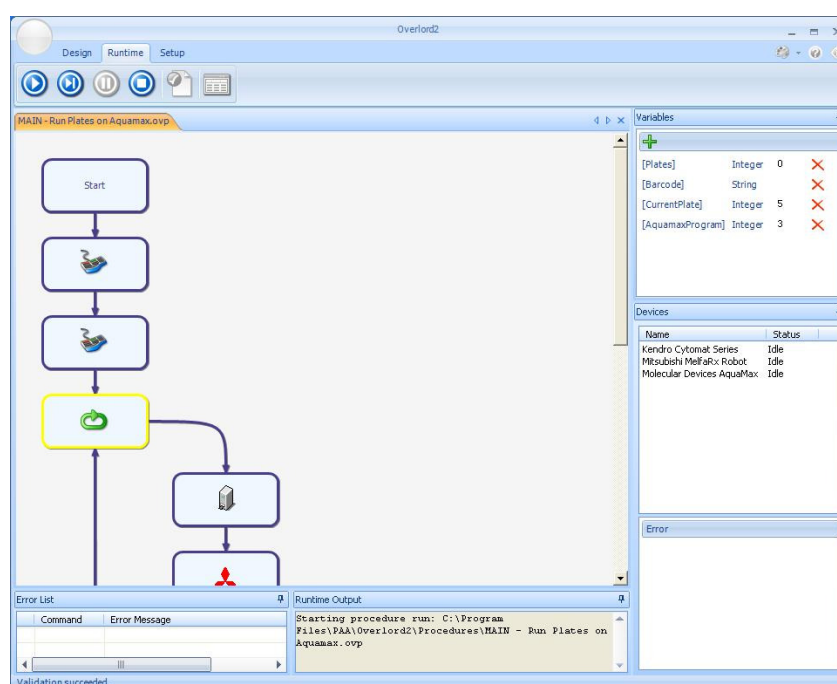
## 7 Running Procedures

### 7.1 Standard (Event-driven) Procedures

Procedures are run from the window that they were created in. This section describes how a procedure can be run and also how it can be validated before a run even begins to ensure it is setup correctly. This section also describes how errors are handled in Overlord3 and how a run is logged.

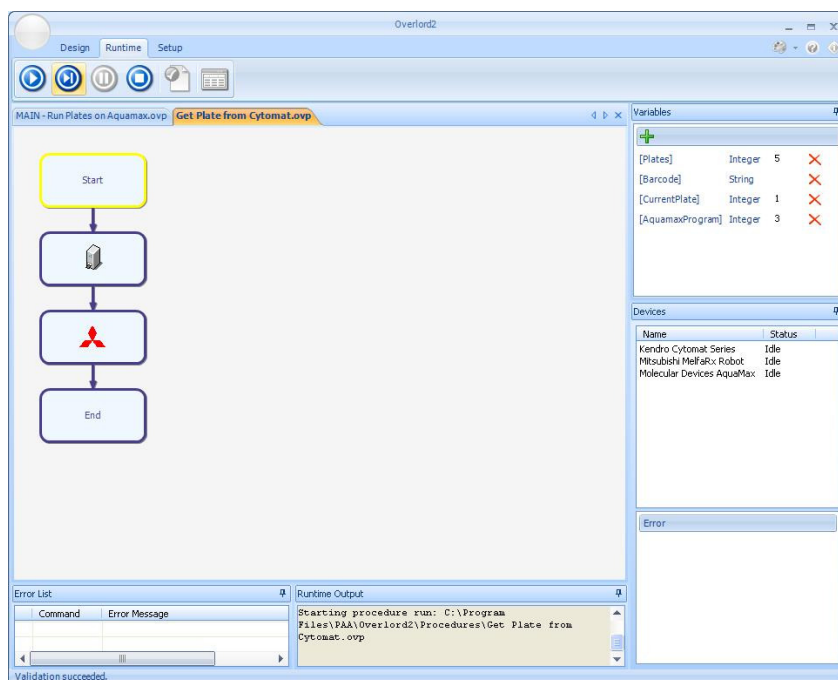
#### 7.1.1 Run

Once a procedure has been created it can be run by simply clicking the Run button on the Ribbon toolbar. The procedure must be saved at this point and a message box will appear asking for the procedure to be saved if it isn't. Before the run starts, the procedure and all of its sub-procedures will be validated to ensure all the Actions can be executed. If this passes, the procedure will begin running).



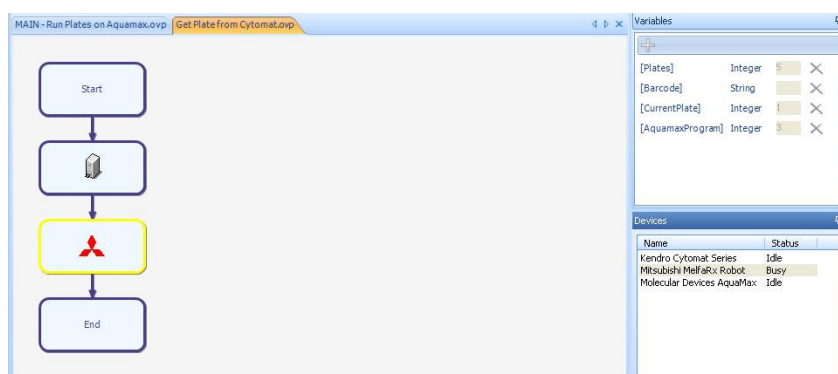
**Overlord3 at Runtime**

At runtime the Command palette that is normally on the left hand side of the screen disappears, but apart from this the screen will appear the same as it does at Edit time. As each Action is highlighted it will be executed and will stay highlighted until the execution is complete. If a sub-procedure is called, this will be loaded onto the screen and execution will continue through the sub-procedure, as seen in Calling a Sub-Procedure at Runtime). The main procedure will stay open at this point and the Call Procedure action that made the call will remain highlighted so it is clear where execution will resume once the sub-procedure has completed (Calling a Sub-Procedure at Runtime).



## Calling a Sub-Procedure at Runtime

When an Action that is running a device, is executed, the status of this device will be updated in the Devices palette that is typically on the right hand side of the screen (Executing a Device Action).



## Executing a Device Action

The run can be paused at any point during the run by clicking the pause button on the ribbon toolbar. Pausing will stop the execution at its current point and will not execute any subsequent actions. If there is an action currently running, or devices that are busy, these will continue running. **Pause** just stops the execution of new actions, it doesn't pause currently running Actions.

To resume the run, click on the Run button on the ribbon toolbar. Alternatively, the stepwise button can be clicked. This will move the execution onto the next Action but will then pause the system again. This is useful when running a procedure for the first time to ensure each Action completes as expected before moving onto the next Action.

The procedure will also pause when it comes up to any Action that has a breakpoint set.

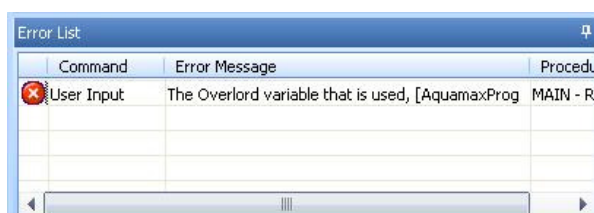
## 7.1.2 Validation

Validation checks that the procedure and all of its sub-procedures contain Actions that will run properly at runtime. Below is a list of just some of the things Validation checks

- Procedures that are being called actually exist
- Variables that are used in an Action actually exist on the system
- That configuration settings have been made for certain devices
- Syntax errors in VBScripts, VB.NET and C# Code Snippets

Validation can be run by clicking on the Validation button in the Runtime section of the Ribbon toolbar. It will also be run automatically when the Run or Stepwise button is clicked.

If there are any issues with any of the Actions in the main procedure or its sub-procedures, these will be listed in the Error List palette that typically is located at the bottom of the window (Validation Error list).



**Validation Error list**

The list explains what procedures and commands have issues and a description for each one of what the problem is. Double clicking on the entry in the list will open that procedure if it isn't already open and highlight the action with the issue.

If there are Validation errors, indicated in the list by the red circle with a white cross, the procedure cannot be run and will not run until the issue is resolved. Entries in the Validation Error list can also be Warnings. Validation warnings are listed alongside a yellow triangle with an exclamation mark inside. Warnings simply alert the user to double check that everything is setup as this action may fail if certain conditions are met. An example would be an Action that uses a device that has a dependency on a certain file existing in a certain folder. The warning for this action would simply tell the user to double-check that when the procedure gets to this action, that key file will exist. This wouldn't be a Validation Error because it maybe that the procedure creates this key file at runtime.

## 7.1.3 Error Handling

Although validation does its best to prevent errors, errors will still come up during a run. These can be due to many different things such as the following.

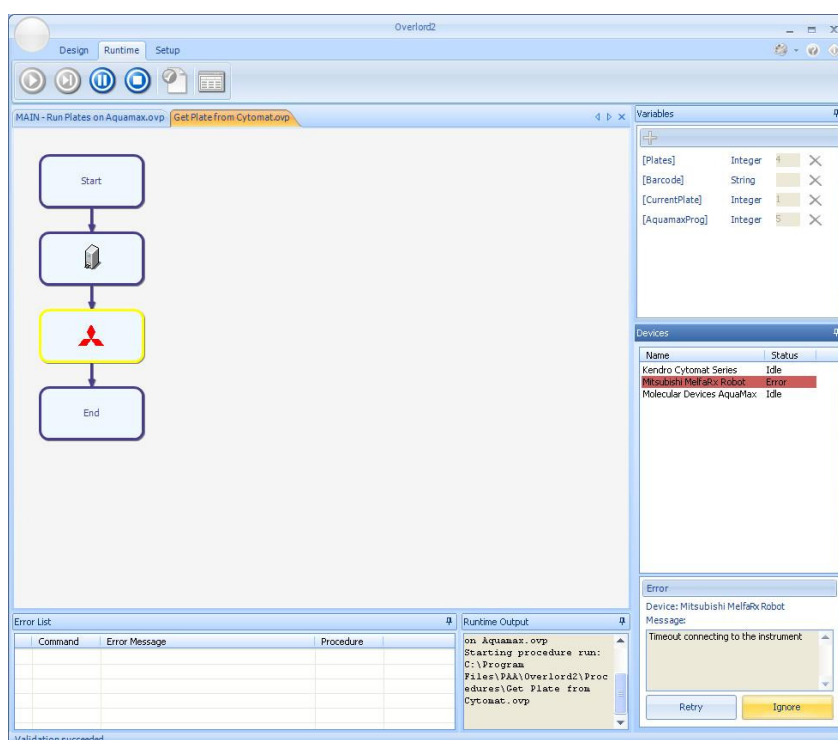
- Faulty instrumentation
- Incorrect correct configuration (e.g. Device setup up on the wrong COM Port)
- Data files not generated in the correct format
- Labware loaded into the wrong position

This list could easily be much longer. There are many different types of error that can happen on any system. Overlord3 will trap all errors that are encountered at runtime and will give the user the chance to respond to the error in three different ways

- **Ignore** – The error will be ignored for this Command

- **Retry** – The Action will be sent to the command again and depending upon the command will either start from the beginning or will resume from where the error occurred
- **Abort** – The entire run will stop

There are 2 different ways that errors are reported and it depends upon what type of command the error is generated in. If the command is a device (e.g. robot, plate reader, pipetting station etc) then the error will be raised in the Device List palette by highlighting in red the relevant error. If this is the only device in error, the Error palette will be filled in with the details of the error. The Retry and Ignore buttons will also become available giving the user the chance to respond to the error as seen in Device Error). If there is more than one device in an error state at the same time, clicking on each device will bring up its error details in the Error palette (Device Error).



## Device Error

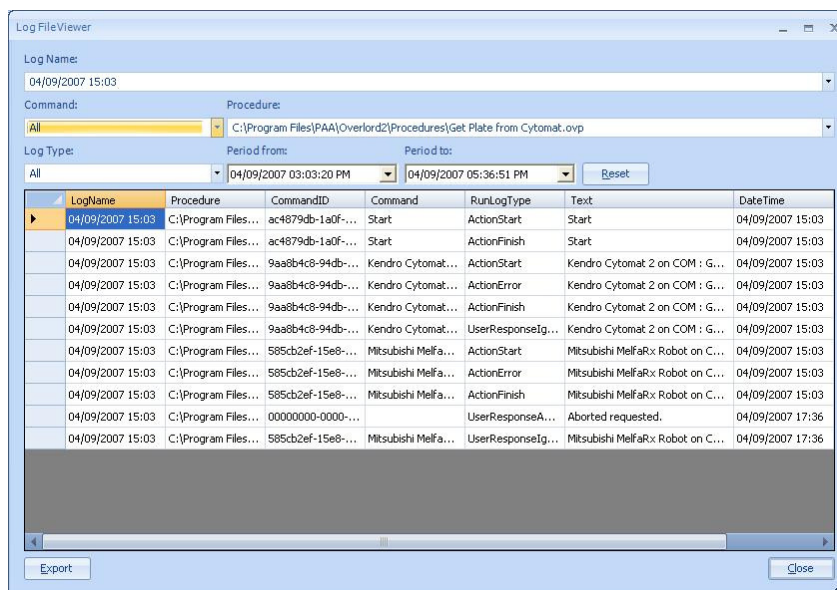
If the command is a non-device, a Flow or Plug-in command (e.g. Message box, Script, Loop etc) then the error message will be brought up as Message box with the Abort, Retry & Ignore options available to the user.

All error responses are recorded into the Logfile.

### 7.1.4 Logfile

Every time a procedure is run, an entry is created in the Overlord Logfile so you can see exactly how the run proceeded (Logfile Viewer). The logfiles are accessed by clicking on the Logfile button in the Runtime section of the Ribbon toolbar.





### Logfile Viewer

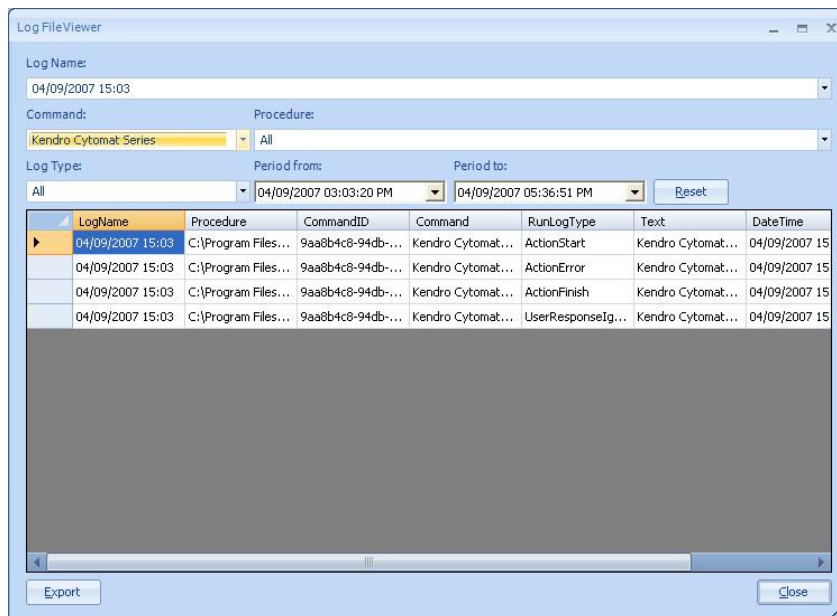
Each time a run starts a new Logname is created using a time/date identifier with the format **<dd/mm/yyyy> <hh:mm>**. Every run that has ever been started will appear in the Log Name drop down list and is listed in reverse chronological order, newest first. Selecting the log you wish to view will populate the Log grid which lists all of the activities which occur during a run. There are different activities that happen during a run, known as RunLogTypes. These are listed below.

Run Log Type	Description
<b>RunStart</b>	When the Run begins
<b>RunFinish</b>	When the Run completes
<b>ActionStart</b>	When an action is first executed
<b>ActionFinish</b>	When the action completes
<b>ActionError</b>	If there is an Error with the Action
<b>UserResponseIgnore</b>	The user has responded to the error be selecting Ignore
<b>UserResponseRetry</b>	The user has responded to the error by selecting Retry
<b>UserResponseAbort</b>	The user has requested the system to Abort

The RunLogType column on the LogGrid sometimes also lists the name of a variable type, Numeric, Text, True/False or Object. This is because the Log also records when variables value changes.

Each entry in the Log is listed with the time that it occurred. There is also an additional column next to this called Timer ticks. This is for trouble shooting systems where there is a timing issue and we need a log recording resolution less than 1 second. This functionality is only for advanced users.

There are various drop down boxes below the Log Name drop down box. These are all filters which allow the Logfile to only display relevant information. For example if the Kendro Cytomat is selected in the Commands Filter drop down box, only entries in the Log that report usage of the Kendro Cytomat will be displayed (Filtered List). This filter can be reset by selecting 'All' in the filter list.



ERROR: stackunderflow  
OFFENDING COMMAND: ~

STACK: