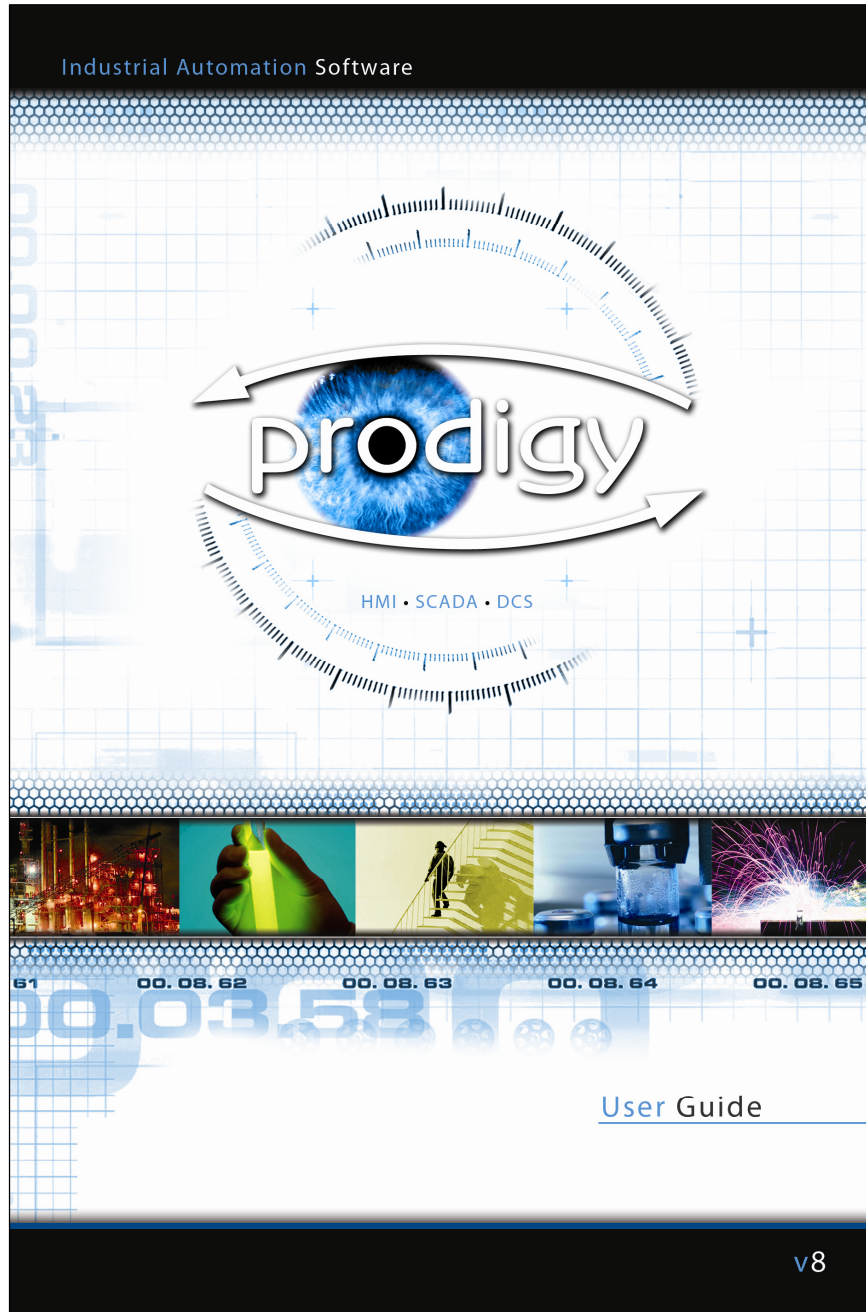


Prodigy User Guide



HMI • SCADA • DCS

Contents

Contents	i
Introduction	17
Limitations of This Manual	17
Getting Started	18
Configuring a Prodigy System	18
Driver Configuration	19
Drivers	19
Driver Configuration	19
Main Buttons	20
Serial Section.....	21
Hardware Handshaking Section	22
Details Section.....	22
Custom Parameters	23
Driver Specific Section	23
Signal Database Configuration	24
The New Age Of Database Configuration	24
What is The Signal Database?.....	24
What Attributes Does A Signal Have?	24
Overview	25
Creating New Signals.....	27
Creating Signals Manually	27
New Signal Wizard	27
Range And Units	27
Driver Address	28
Alarms	28
After Completion	29
Creating Many Signals At Once.....	29
Creating Signals By Duplicating Existing Ones.....	29
Manual	30
Substitution	31
Examples Of Substitution Strings	32
Importing Signals From Other Systems	32
Adding Interconnect Signals	33
Increasing The Database Size.....	34
Changing The Attributes Of A Signal	36
Data Entry Basics	36
Field By Field In The Signal Database.....	37
Index.....	37
Signal	37
Description	37
Signal Type	37
Driver	38

Address	39
Hex Address	40
Update Rate	40
Range.....	41
Units	41
Event Text	41
Groups	42
Reset	43
Record	44
Alarm.....	45
VL Alarm Level, L Alarm Level, H Alarm Level & VH Alarm Level.....	48
VL Alarm Txt, L Alarm Txt, Normal Alarm Txt, H Alarm Txt & VH Alarm Txt .	48
VL Alarm Priority, L Alarm Priority, H Alarm Priority & VH Alarm Priority.....	48
Alarm Nag and Comments	48
Alarm Deviation	48
Alarm Recording	49
Linear Interpolation	49
End A Raw Value	49
End A Scaled Value.....	49
End B Raw Value	49
End B Scaled Value.....	50
Copy To.....	50
Signal Smoothing	50
Smoothing % & Smoothing Reset.....	51
Invert Digital	51
Black Box Slot.....	51
Black Box Channel.....	51
Copying The Attributes Of One Signal To Another	52
Deleting Signals.....	52
Navigating The Signal Database – Filtering, Searching, Etc.....	53
Column Presets	53
Signal Name Search/Filter	54
Filtering By Signal Attributes.....	55
Filtering By Selection	58
Saving The Signal Database	59
Simply Saving.....	59
Save And “Go Live”	59
Selective “Go Live”	60
Disaster Recovery.....	61
Loading From The History File	61
Loading From AutoSave.....	61

Signal Database Import/Export 63

How to use the Import/Export facility	63
Create the Initial Database.....	63
Export the Signal Database.....	63
Import the Database Back Into Prodigy.....	63
Friendly Addressing	63
Modbus.....	63
MelsecA.....	64

Signal Monitor 65

Signal Monitor.....	65
Navigating – Filtering, Searching, Etc.....	65
Altering Signal Properties.....	65
Sorting	67
Printing	67

Display Builder 68

What Display Builder Can Do.....	68
How Do You Construct A Display Builder Display?.....	68
The Drawing Toolbar	69
Selection.....	69
Point Edit.....	69
Zoom	69
Line(s).....	69
Curve(s).....	69
Rectangle.....	70
Rounded Rectangle.....	70
Ellipse.....	70
Closed Polygon.....	70
Text	70
Button.....	71
Bitmap.....	71
Trend	71
Heatmap.....	71
Advanced Heatmap	71
Heatmap Extra.....	71
Grid	72
Frame	72
Dial.....	72
Bar Graph.....	72
Intelligent Objects.....	72
Forms.....	73
How Do You Place an Object?.....	74
How Do You Move Or Resize An Object?	74
Orientation Techniques	74
The Undo and Redo Commands.....	76
How Do You Define the Object's Properties?	76
What Makes Display Builder Objects Dynamic?	77
General Object Properties.....	77
Visibility.....	77
% Fill.....	78
Fill Colour	79
Line Attributes	80
Size.....	81
Position.....	82
Orientation	83
Button.....	84
Special Button Properties	91
On-line Media	93
File Viewers	94
Slider	96
More Specialised Properties	96
Grid	96
Frame	98
Text Colour	99
Value Display.....	100
Bitmap.....	103
Trends & Trend Properties	104
Trace Properties	108
Alias Tag Properties	109
Profile Curve Properties.....	109
Dials	111
Dial Properties.....	111
More Dial Properties	113
Other Tabs On The Dial Property Page.....	114

Bar Graphs.....	114
Frames	115
ToolTips	116
Animation Timers.....	117
Making A Better Display.....	118
Dynamic Tag Substitution.....	118
Making Composite Objects	118
Grouping Objects.....	119
Ungrouping Objects.....	119
Grouped Attribute Changes	119
Duplicating Objects	120
Aligning Objects.....	120
Spacing Objects	121
Sizing Objects.....	122
Changing the Z-Order.....	123
Setting Defaults	123
Altering Refresh Rates.....	124
The Status Bar	125
Finding and Replacing Tags Used in a Display.....	126
Changing the Tags Used in an Object	126
Changing Tags in Multiple Objects	127
Finding out Where Tags Are Used	127
Saving Objects to the Object Library	128
SuperObjects	128
Loading Objects from the Object Library.....	129
Deleting Objects	130
Saving a Display.....	130
SuperObjects.....	131
Using A SuperObject.....	132
Activating the New Signals	133
Interaction with Database Configuration.....	133

Trending **134**

Live Trending	134
Axis Sliding	134
Zooming	135
Graticules.....	136
The Trace Menu.....	136
The Global Menu.....	137
Historical Trending.....	138
Archived Data.....	139
Alias Tags.....	139
The Parts of the Historical Trend Display	139
Graph Window (1).....	139
Cursor Time (2).....	139
Cursor (3).....	140
Information Window (4).....	140
Axis Span Information (5).....	140
Historical Trend Toolbar (6).....	140
Y-Axes (7).....	140
X-Axis (8).....	140
The Historical Trend Toolbar	140
Back.....	140
Data Location	141
Print	141
Title	141
Select Signals	142
Align Curve	142
Peak Mode.....	142

Select Batch.....	143
End Batch.....	144
Relative Time.....	144
Time Span.....	145
Select Date/Time.....	145
Comments (Notes).....	146
Zooming.....	146
Undo Zoom.....	147
Fit to Axes.....	147
Less Smoothing.....	147
More Smoothing.....	147
Multi Trending.....	147
Show “A” Trend.....	148
Show “B” Trend.....	149
Link Trends.....	149
Split Trend Display.....	150
Previous Period.....	150
Previous Value.....	150
Large Left Scroll.....	150
Small Left Scroll.....	151
Centre Cursor.....	151
Small Right Scroll/Large Right Scroll/Next Value/Next Period.....	151
Get New Values.....	151
CSV Data Export.....	151
SPC.....	153
Help.....	153
Zooming, Dragging Etc.....	153
Data Value Read Off.....	153
Digital and Finite State Signals.....	153

Menu Bar 155

What Is the Menu Bar?.....	155
Customising the Prodigy Menu Bar.....	155
The Navigation Tab.....	156
Adding Menu Items.....	157
Defining Properties for Menu Items.....	158
Menu Item Properties.....	158
Directory Based Menu Item Properties.....	159
Program Item Properties.....	161
The Appearance Tab.....	162
The Security Tab.....	165
Properties Protection.....	165
Impact of Security on Menu Bar Appearance.....	166

Alarms 167

What Is an Alarm?.....	167
The Alarm Display.....	167
What Appears In the Alarm Display.....	167
Accepting Alarms.....	168
Alarm Nag.....	168
Alarm Commenting.....	169
Security Considerations.....	169
Alarm Display Properties.....	169
Full Screen Mode.....	172
Selecting Columns For The Full Screen Display.....	174
Applying Filters To The Full Screen Display.....	174
Selecting Alarms.....	175
Events.....	175
Alarm Recording.....	175

SMS & Email Messaging	176
SMS Active Alarms.....	178
Configuring SMS Services	179
Selecting The Service Type.....	179
Configuring A Standard Modem	179
Configuring The Radio Modem	180
Interactive Facilities With The Radio Modem.....	181
Configuring Email Services.....	182
Interactive Facilities With Email	183
Alarm History Viewer	184
Signal Filtering/Highlighting.....	185
Display Launcher	187
Display Launcher.....	187
Display Launcher Configuration	187
Control Panel	189
A Word About Services.....	189
What Does Prodigy Service Start	189
Access to Prodigy Services.....	190
Prodigy Control Panel.....	191
Service Status Tab	191
Optional Services Configuration	192
XForms Tab.....	194
Additional Programs Tab.....	195
Watchdog Tab	196
Dual Computer Tab	197
Security Tab	199
Lockdown Tab.....	201
Default File Locations Tab	201
Web Tab	202
Application Monitoring	202
Actions Of The Watchdog.....	202
Watchdog Driver	203
Deferred Reboot	203
Stopping the Services	203
Error Logging	203
Quick Start	204
Quick Start.....	204
Security	205
How Security Works	205
User Management.....	205
User Details	206
Security.....	207
Privileges	207
The Default User	208
Logging On.....	209
Logging Off.....	209
Prodigy Screen Saver	210
Screen Saver & Auto Log-off.....	210
Audit Trail	212
Audit Trail	212

Filtering	213
Sorting	214
Printing	214
SigScan	215
What is SigScan	215
Using SigScan	215
Name	216
Description	216
Type	216
Direction	217
Recorded	217
Deleted	217
Groups	217
Sources	218
Filter Button	218
Signals Not In the Database	218
Alias Tags	219
What Are Alias Tags?	219
How Do Alias Tags Work	219
Defining Alias Tags	219
Assigning Real Tags To Alias Tags	220
Using Alias Tags	221
Signal Disabler	222
Signal Disabler	222
Display Replay	223
What Is Display Replay?	223
Using Display Replay	223
XForms	225
What are XForms?	225
What are Recipes?	225
What are Batches?	225
A Word About Opening Databases	225
Exceptions	226
Launching XForm Edit	226
The XForm Editor	227
The XForm Toolbar	227
Creating an XForm Dialog	227
XForm Form Properties	227
Adding Objects	230
Pointer	230
Label	230
Edit Box	230
Check Box	231
Combo Box	231
Button	231
Grid Control	231
Date/Time	231
Picture Box	231
Radio Button	231
Group Box	232
Morph	232

Properties of XForm Objects	233
Source/Destination	233
Appearance	235
Numeric	237
Text.....	238
Date/Time	240
List Items	241
Buttons.....	242
Primary Database.....	244
Check Boxes.....	245
Radio Button.....	245
Picture.....	246
Data Grids.....	246
Grid/Column Properties.....	247
Secondary Database.....	249
Show and Auto Label	250
Show	250
Auto Label.....	251
Positioning and Formatting Objects	251
Aligning Objects.....	251
Sizing Objects.....	252
Spacing Objects.....	252
Guide Lines and Magnetic Guide Lines	252
Duplicating Objects	253
Set Tab Order	253
Detach Group Box.....	253
Tag Replace.....	254
XForm Design Requirements	255
Selecting a Batch	255
Showing the Selected Batch on a Display	255
Starting a Batch	255
Ending a Batch.....	255
Recipes	255
Using an XForm	255
An Example XForm.....	256
XForms as Reports	257
The Batch and Recipe Databases.....	257
Batch Chooser Configuration	258
The Batch Chooser	258
Comments	261
Comments.....	261
Profile Curves	263
Purpose of Profile Curves.....	263
Preparing A Profile Curve	263
The Profile Editor	263
The Profile Toolbar	264
Screen Properties	265
Defining A New Profile Curve	266
Modifying a Curve.....	266
Zooming	267
Rescaling	267
Scrolling	267
Point Properties & Annotation.....	267
Comment	268
The CSV File.....	268

Using the Finished Profile Curve	269
Exporting Curves From A Trend.....	270

Easy Report 271

Using Easy Report.....	271
The Tool Bars.....	271
Main Toolbar.....	271
View Toolbar	272
Arrange Toolbar.....	273
Page Toolbar	274
Tool Toolbar.....	275
Object Properties.....	276
Text Properties	276
Bitmap Properties.....	278
Report Properties.....	278
Configuring A Data Block Object.....	280
Data Table Properties Tab.....	280
Table.....	280
Headings.....	280
Statistics.....	280
Column list.....	280
Column.....	280
Field.....	281
Calculated.....	281
Heading.....	281
Width.....	281
Use description table.....	281
Group Heading.....	282
Sort.....	282
Colour.....	282
Justify.....	282
Indent.....	282
Date/Time Tab.....	282
Numeric Tab.....	283
Boolean Tab.....	283
Colour Tab.....	284
Statistics Tab.....	284
Grid Tab.....	284
Font Tab.....	285
SQL Selection Tab.....	285
Configuring A Field Object.....	286
Date/Time Tab.....	286
Numeric Tab.....	287
Boolean Tab.....	287
Colour Tab.....	287
Configuring A Chart Object.....	288
Bar Chart Properties.....	288
Chart Type.....	288
Legend (X Axis).....	289
Y Axis.....	289
Title.....	290
Grid Lines.....	290
Background Colour.....	291
Data Values.....	291
Colours.....	291
Other Options.....	291
Bar Chart Items List.....	292
Table.....	292
Field List.....	293

Field Item	293
Legend	293
Statistics	293
Colours	294
Other Styles Of Barchart	295
Statistical	295
Series	295
Stacked	296
Stacked 100%	296
Simple Pie	297
Statistical Pie	297
Series Pie	298
Getting Started With Easy Report	299
Create a blank report	299
Set the report properties	300
Place headings and footers	300
Place a data block	301
Email Reports	304

Proof Of Process Report 305

Proof Of Process Reports	305
Configuring A Proof Of Process Report	305
Select A Database	307
Placing Objects	307
Object Properties	309
Text Object	309
Database Field	310
Alarm List	310
Trend	311
Running A Proof Of Process Report	312

Multigraph 314

X-Y Plots	314
Database Considerations	314
Using Multigraph	314
Getting Started	316
Plot	316
New Graph	317
X-axis	318
Y-axis	319
Header	321
Fixed Text	322
Data Text	322
Other Functions	323
Zoom	323
Full Size	323
Outline View	323
Duplicate	324
Delete	324
Alignment	324
Hierarchy	325
Run	325
Run Multiple	325
Multigraph Selection	326

Statistical Process Control (SPC) 327

An Overview Of SPC	327
The Prodigy SPC Facility	327

The Parts Of The SPC Program.....	328
The Main Display	328
The Toolbars	328
The Standard Toolbar.....	328
The Charts Toolbar	329
Using the SPC Program.....	330
Creating A Project.....	330
Project Properties	331
Signals.....	331
Data Range.....	333
Chart Settings.....	334
More Chart Settings	336
Run Rules.....	337
Specification Limits	338
Creating A Run.....	338
CSV Data	339
Manual Data.....	340
Setting Up Colours.....	340
Viewing The Data In Prodigy SPC.....	341
The Data View	341
The Distribution Histogram.....	342
The Average & Range Chart	343
Machine Capability v Process Capability.....	344
The Median & Range Chart	345
The Individual & Range Chart	346
SPC In Conjunction With Historical Trends	347

The Alarm Scheduler 348

The Purpose of the Alarm Scheduler.....	348
How to Use the Alarm Scheduler.....	348
Preparing the Driver	348
Preparing the Signal Database.....	348
AutoAssign.....	348
Alarm Schedule Signal.....	348
Creating Alarm Schedules.....	349
Selecting Alarm Schedules.....	349

Access Database Configuration 350

Use Of Access Databases	350
The Database Configuration Program	350
Main Dialog.....	350
Database Definition Dialog	351
Tables	351
Fields	352

Batch Data Recorder 354

Purpose Of Batch Data Recorder (BDR).....	354
BDR Configuration Program.....	354
Configuration Properties	355
Main	355
Run Parameters.	355
Reports	357
Start Condition	358
End Condition	359
Start Reset	360
End Reset	360
Start Data.....	361
Continuous Data.....	362

End Data	363
Alias.....	365
Signal Attributes.....	366
BDR in Action.....	367
BDR in Conjunction with XForm.....	367
Automated Test Sequencer (ATS)	368
What Is ATS?	368
Configuration.....	368
Configuration Browser	368
Global Options.....	369
Signals	369
Values.....	371
Ramps	371
Triggers.....	372
Counters.....	373
Alarm Schedules.....	373
End Conditions	374
SLANG.....	374
Down Time Monitoring	375
Overview Of Down Time Monitoring.....	375
Down Time Logger Configuration	375
Configuration Selection.....	375
Configuring Properties.....	377
Advanced Properties.....	378
Short Codes	380
Down Time Logger	381
Typical Down Time Scenarios	382
Down Time Reason Configuration.....	383
Down Time Reason Configuration Program	383
Down Time Logger Reason.....	386
Down Time Viewer	388
The Display Area Explained.....	388
Date Range	388
Duration.....	389
Event Type	389
Levels	389
Comment	390
Filtering	390
Filtering by Configuration.....	390
Filtering by Event & Levels	390
Filtering by Duration	391
Filtering by Selection	392
Selecting A Date Range.....	392
Stepping Through The Data.....	392
Editing Data.....	393
Edit Single Record.....	393
Combine With Previous/Next Record	393
Split Record.....	393
Reporting & Exporting Data.....	394
Reporting	394
Down Time Overview	395
The Tool Bar.....	395
Preferences	397
Time Spans.....	398
Display.....	398
Item Height.....	398
Refresh Minutes.....	398

Data Processing	399
Data Processor Configuration	399
Configurations	399
Properties	400
Selection Tab	400
Processing Tab	402
Down Time Reporting	403
Data Processing / Report Configuration	404
Running A Down Time Report	404

Shop Floor Terminals 405

What Are Shop Floor Terminals?	405
Shop Floor Terminal Configuration Program	406
Device Configuration	407
None	407
Toggle Description	407
Cycle Machine	408
Cycle Signal	408
Cycle Alarm	409
Down Time Reason	410

SLANG (Sequence LANGUAGE) 411

What Is SLANG?	411
The SLANG Language	411
Variables	411
Types	411
Scope	412
Indirection	413
Constants	413
Macros	413
Statements	414
Set	414
Turn	414
Lock, Unlock	414
Wait	415
If Else Endif	415
Repeat	416
Goto, Label	417
Routine, EndRoutine, Return, Call	417
Global Routines	419
Include Files	422
Stop, Start	423
Run	423
Comment	423
Inline	423
Pause	423
Signal Attributes	424
Expressions	424
Conditions	424
The SLANG Editor	426
Creating A New Program	427
Adding Program Elements (Statements)	428
Comment	429
Include	430
Label	430
Goto	430
Set	431
Variable Selection	432
Turn	432

Lock.....	433
Unlock.....	433
Wait.....	434
If.....	434
Else.....	435
Repeat.....	436
Call.....	437
Routine.....	437
Return.....	437
Run.....	438
Stop.....	439
Start.....	440
Modifying A Program.....	440
Selecting Programs.....	440
Saving Programs.....	440
Deleting Programs.....	441
Exiting From The Editor.....	441
Abandoning Changes.....	441
Adding Statements.....	441
Modifying Statements.....	441
Deleting Statements.....	441
Multiple Deletion.....	442
Copying & Pasting.....	442
Cutting & Pasting.....	442
Find.....	443
Display Properties.....	443
SLANG Control Panel.....	444
The SLANG Runtime Environment.....	444
The SLANG Control Panel.....	445
The Variables Dialog.....	447
The Array Variable Dialog.....	448
Debugging SLANG Programs.....	450

OPC 452

Introduction to OPC.....	452
Servers Groups and Items.....	452
The OPC Software.....	452
Using OPC to Communicate between Prodigy and Hardware Modules.....	453
Mapping Prodigy signals onto OPC Items.....	453
Connecting to OPC Servers.....	454
Setting a Default Server.....	456
Filtering.....	456
Selecting an OPC Item.....	457
The OPC Driver.....	458
The OPC Server.....	459
OPC Server Configuration.....	459
Diagnostic Tools.....	461
Using the Standalone OPC Client.....	461

DDE 466

DDE.....	466
Prodigy DDE Server.....	466
Prodigy DDE Client.....	467

Controllers 468

What is a Controller.....	468
How PID Control Works.....	468
Configuring the Controllers.....	470

Adding, Removing, Starting and Stopping.....	470
Setting the Parameters	471
Applying or Cancelling.....	473
Auto Tuning a Controller	473
The Auto Tune Parameters	474
Performing an Auto Tune.....	475
Manually Modifying the P, I and D Gains.	476
Timed Reports	478
What Are Timed Reports	478
Timed Report Configuration	478
Calibration	481
Calibration.....	481
Creating a Configuration File.....	482
Calibrating Signals	484
Calibrating Inputs.....	485
Calibrating Outputs	485
Calibrating Multiple Inputs and Outputs.....	485
Calibration Check.....	486
Calibration Reports	486
Signal Event Logger	487
Signal Event Logger.....	487
Signal Event Viewer.....	487
Black Box Recorder	490
Overview of Black Box Recorder	490
Using Black Box Recording.....	490
Setting Up The Driver	490
Configuring Signals To Be Recorded.....	491
SLANG Extensions for Black Box Recorder.....	492
BBLoggingRate.....	492
BBDumpToDatabase	493
Expert Data System	494
What Is The Expert Data System?.....	494
Configuring the Expert Data System.....	494
Batch Properties	495
Decision Items.....	495
Action Items	499
Deleting	500
Running An Analysis	500
Automatic Archiving	502
Automatic Archiving.....	502
Archive	502
Databases	503
Incremental Only.....	503
Source Data Deletion Criteria	503
Directory	504
Data Criteria	504
Archive Destination.....	504
Archive Time	505
Configuration Criteria	506
Archive Destination.....	506

Archive Time.....	507
History Folder.....	507
SpaceWatch Criteria.....	508
HTML Reports and Web Upload	509
Introduction	509
Producing HTML Reports	509
While viewing a report on the screen	509
Automatically Using the Timed Reports Facility.....	509
Automatically Generating a Crystal Report at the end of a batch	510
Uploading HTML files to a Website	510
Configuring the Details for Uploading Files	510
Uploading Files	511
Dual Computer Support	512
What is Dual Computer Support?.....	512
Setting up Dual Computer Support.....	512
Preparing Your PCs	512
Configuring Dual Support	512
How Best To Approach Dual Support.....	512
Configuring Options	513
When A Fault Occurs	513
Prodigy Workstation	514
What Is Prodigy Workstation	514
How Prodigy Workstation Works.....	514
Limitations.....	514
Using Prodigy Workstation	514
Software Installation.....	514
Licensing Requirements	514
Security.....	515
System Preparation	515
Running Prodigy Workstation	516
Adding A New Machine.....	517
The Menu Bar.....	518
Prodigy Web Viewer	519
What Is Prodigy Web Viewer.....	519
Web Viewer Configuration.....	519
Web Tab	519
User Configuration	521
Steps to Set Up and Connect to Prodigy Web Viewer.....	521
Appendices	523
Drivers	523
SLANG Extensions	523
System Administration Tasks	523
Run Time Parameters	523
Communications Wiring.....	523
Index	525

Limitations of This Manual

This manual is revision 20090114/8.0

This manual is not intended to be a tutorial, nor does it replace proper formal training. The facilities described in this manual are those available with the full Prodigy system, known as “*Complete Development*”. Some facilities are not available with other versions of the product, such as *Lite* or *Runtime*. See the “**Help About**” for details of your license or contact your supplier.

The manual describes Prodigy in use on Windows Vista. While Prodigy itself should not vary between operating systems, some slight variations in operating system procedure may occur on some other systems (95, 98, 2000, NT and XP). This particularly relates to such things as Services (there is no such facility on 95 and 98).

Configuring a Prodigy System

- If your system is interacting with real-world plant I/O then the first thing to do is to use **Driver Configuration** to set up the correct drivers for the I/O. Add a driver for each I/O loop that there will be, set up its communication parameters and set it to **Auto Start**. If there is no I/O then just proceed to the next stage.
- Next use **Signal Database Configuration** to create some signals. Associate those signals that access any I/O with the drivers you have just added. Remember to set an **Update Period** (this is the frequency with which they are polled) for any I/O signals, otherwise no polling will take place.
- Hit the 'Go Live' button this is a necessary step to make any new signals become "live".
- Use **Signal Monitor** to check the signal values and update times of the signals that you have added. The update times should change roughly in accordance with the **Update Period** that you have set in **Signal Database Configuration**.
- Create some displays in **Display Builder**.
- Write any **SLANG** that may be required.

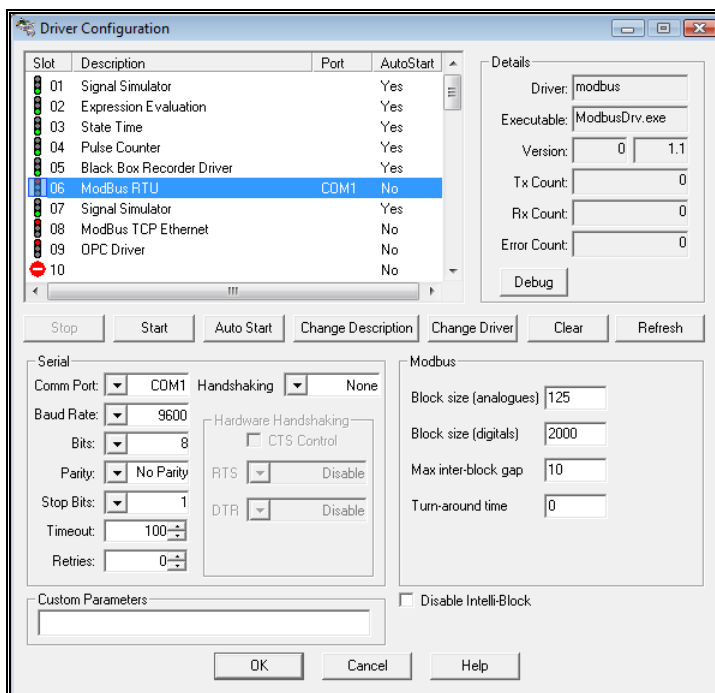
There is a document on the Prodigy CD called "Getting Started With Prodigy" which covers this in greater detail. If you are not familiar with Prodigy it is highly recommended that you read this before proceeding.

Driver Configuration

Drivers

There are two types of driver within Prodigy - those that process data from outside sources (such as hardware devices) and those that process data from signals within Prodigy to provide values for other signals (such as **Expression Evaluation**). The **Driver Configuration** program defines which drivers Prodigy runs, along with any necessary configuration information. Additionally, this program allows new drivers to be added and also provides some feedback information from drivers that are running.

Driver Configuration



The Driver Configuration dialog consists of a scrolling list that contains all of the drivers currently installed for a Prodigy installation. Note that a driver can be installed in any of the slots available, so for instance in our example above, there may be a driver installed for slot 15, which is not currently visible. Do not assume that all drivers that are installed appear at the top of the list, though it is good practice to install drivers in the first available slot and not leave unnecessary gaps.

Each line of the scrolling list has four parts to it:

- **Slot.**
The logical location at which the driver resides.
- **Description.**
The description of the driver at this location.
- **Port.**
The communications port on the computer through which this driver will communicate with any external devices.
- **AutoStart.**
Indicates whether or not this driver will be started by Prodigy when the Services start (see the relevant section on Services).

By clicking on a **Slot** the information for the driver in that slot will be displayed in the other fields on the dialog. This information varies with the type of driver and not all fields will be used. Those fields that are not relevant will be greyed out and therefore inaccessible. Some fields (particularly those in the **Details** section) will be read-only. Some buttons may be greyed out to indicate that the action they control is not available at

the present time. For example, if a driver is not running the **Stop** button will not be accessible but the **Start** button will, and vice-versa.

The configuration section in the bottom right corner of the dialog changes according to what type of driver is selected and contains very specific configuration information that is explained later in this manual.

Main Buttons

The actions performed by the buttons below the scrolling list area are as follows:

- **Stop.**
This will stop the currently selected driver. The Driver Configuration program communicates with the Prodigy Service and requests that it stops the driver. This is the only appropriate way to stop a driver that is running. If you use the Windows™ Task Manager to stop the driver the Prodigy Service will simply restart it. When the Service has succeeded in stopping the driver the **Stop** button will become inactive and the **Start** button will become active.
- **Start.**
This will attempt to start the currently selected driver. The Driver Configuration program communicates with the Prodigy Service and requests that it starts the driver. When the Service has succeeded in starting the driver the **Start** button will become inactive and the **Stop** button will become active. If the driver fails to start the button states will not change.

*Note that when a driver is started any changes that you may have made to its configuration will be taken into account. If the configuration is severely flawed it may cause the driver to terminate (the driver will issue a warning dialog first). The Service will attempt to restart the driver and it will fail again. This cycle will be repeated a maximum of six times before the Service stops attempting to restart it. It would be a good idea to use the **Stop** button on the Driver Configuration dialog to stop the driver so that Prodigy Services does not keep attempting to restart it.*

- **AutoStart.**
This determines whether or not the driver is started automatically when the Prodigy Services start up. Pressing this button toggles the state of **AutoStart**.

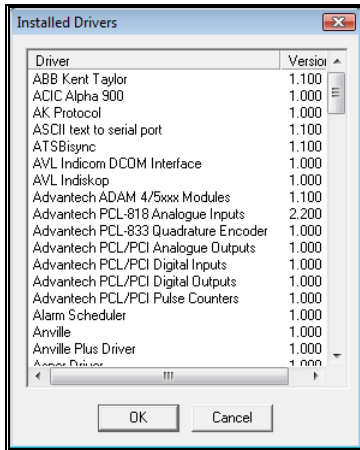
*Note that stopping or starting the driver using the **Stop** and **Start** buttons described earlier does not affect the **AutoStart** state. Stopping a driver but leaving its AutoStart set to **Yes** means that next time the Services are started the driver will be started again.*

- **Change Description.**
This changes the description for the currently selected driver. When the driver is first installed it will have a generic description (simply saying what type of driver it is). You may want to alter this so that it describes not only the driver but also gives some information as to what it is controlling or which loop it is on. This will be useful if there are several drivers of the same type attached to different I/O ports, so that when configuring signals in **Signal Database Configuration** it is apparent which driver is which.

- **Change Driver.**

This allows a new driver to be selected for the currently selected slot. A dialog is displayed listing all of the currently installed drivers, this dialog also providing the option of loading a new driver from an installation disk.

To select one of the drivers from this list simply click on it and then press the **OK** button.



When you have selected the relevant driver from the **Installed Drivers** dialog it will be located in the currently selected slot, replacing any driver that was already at that slot.

- **Clear.**

This removes the driver from the currently selected slot (the slot then becomes vacant).

- **Refresh.**

This re-loads all driver information.

Serial Section

This section of the **Driver Configuration** dialog covers configuration of drivers that operate via serial communications. It is advisable that you are familiar with communications protocols and terminology if you intend to change the driver configuration.

When you load a driver into a slot a default set of communications parameters are loaded with it. These will usually work for the selected device type, though the com port will need to be selected and the baud rate may need to be altered.

The fields within the serial section are as follows:

- **Comm Port.**

This is a drop-down list containing communication ports 1 to 99. Select the relevant com port for this driver, being aware that it is your responsibility to ensure that that com port actually exists on your computer and that the hardware is connected to the correct port.

- **Baud Rate.**

Select the required baud rate from the list presented here.

- **Bits.**

Select the required number of data bits from this list.

- **Parity.**

Select the required parity from this list.

- **Stop Bits.**

Select the required number of stop bits from this list.

- **Timeout.**

Enter the timeout period in milliseconds. This specifies the length of time that the driver will wait for a response from the hardware before it decides that no response is forthcoming.

- **Handshaking.**

Select the relevant handshaking protocol to be used, either **None**, **Xon/Xoff** or **Hardware**. If hardware handshaking is selected the section on **Hardware Handshaking** becomes accessible.

Hardware Handshaking Section

This allows the parameters defining hardware handshaking to be configured.

- **CTS** (clear-to-send) Control.
If this box is checked the CTS line is monitored to determine if the driver can send characters to the device.
- **RTS** (request-to-send).
The method used to tell the device it is OK to send characters to the driver.
 - **Disable.**
Disables the RTS line when the device is opened and leaves it disabled.
 - **Enable.**
Enables the RTS line when the device is opened and leaves it on.
 - **Handshake.**
Enables RTS handshaking. The driver raises the RTS line when the input buffer is less than half full and lowers it when the input buffer is more than three-quarters full.
 - **Toggle.**
Specifies that the RTS line will be high if bytes are available for transmission. When all buffered bytes have been sent, the RTS line will be low.
- **DTR** (Data-Terminal-Ready).
The method used to indicate to the device that the computer is ready for transmission of data.
 - **Disable.**
Disables the DTR line when the device is opened and leaves it disabled.
 - **Enable.**
Enables the DTR line when the device is opened and leaves it on.
 - **Handshake.**
Enables DTR handshaking. The driver raises the DTR line when the input buffer is less than half full and lowers it when the input buffer is more than three-quarters full.

Details Section

This part of the dialog contains feedback information from the driver in the currently selected slot. The Tx Count, Rx Count and Error Count fields are particularly useful in determining if the driver is communicating properly with the device (but **note** that signals must have been configured to use this device before any communications will be attempted with it).

- **Driver.**
The name of the driver that is selected for this slot.
- **Executable.**
The actual name of the executable program that forms the driver for this slot.
- **Version.**
The version number of the driver at this slot.
- **Tx Count.**
The number of messages that the driver has transmitted since the driver was last started.
- **Rx Count.**
The number of messages that the driver has received since the driver was last started.
- **Error Count.**
The number of errors that have occurred whilst processing messages from the device. Examples of errors that can occur are "no message received" (device did not respond) or "invalid message received" (message may have been garbled due to communications error).
- **Debug.**
Pressing this button will cause the currently selected driver to begin generating diagnostic messages that can be used to help debug a driver problem. It is not really an end-user facility, but you may be asked to use it by your supplier or integrator.

Custom Parameters

Some drivers require custom parameters, and these are entered here. Details of such parameters are given with the relevant driver information later in this manual.

Driver Specific Section

This section occupies the bottom right quadrant of the screen, and in the screenshot earlier it is labelled **Modbus**. The number and types of fields in this section vary according to the driver that has been selected. Details are given later in this manual.

Signal Database Configuration

The New Age Of Database Configuration

In versions of Prodigy prior to 7.3 the Signal Database Configuration program was somewhat cumbersome and unintuitive. Amongst its many nuances was the fact that after signals had been added to the signal database, or if almost any kind of changes were made other than the most basic, Prodigy needed to be restarted in order for those changes to take effect. Additionally, it was impossible to create a block of similar signals or to alter the attributes of several signals at once. The only way to achieve this kind of editing in the past was to export the signal database to CSV, make the changes in Excel™ and then re-import the signal database.

Prodigy Signal Database Configuration now gives you spreadsheet-style editing that allows you to create, duplicate or modify whole blocks of signals in one go. Additionally, all changes that are made to the signal database can be applied to the live (active) database immediately with no need to restart Prodigy, even the creation of new signals.

Prodigy Signal Database Configuration also doubles as Signal Monitor and you can switch directly from editing to monitoring view to check that your changes are working as expected.

What is The Signal Database?

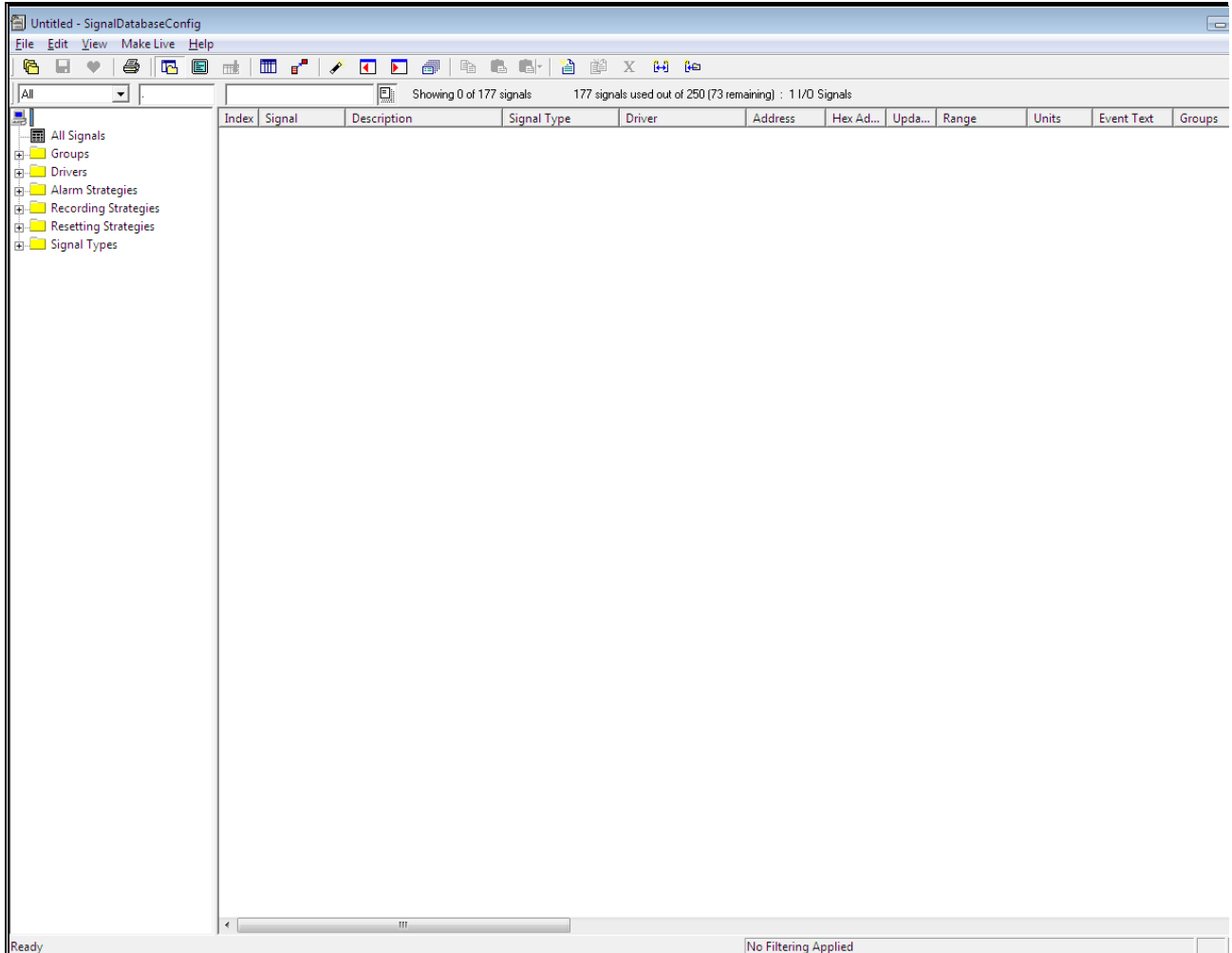
The signal database is at the heart of the Prodigy system. It contains all of the configuration information about the signals being monitored, along with current state information. An *event driven* process called the **Signal Manager** updates the Signal Database in real-time whenever a signal value on the plant changes and also informs all other interested processes that a change has occurred so that they can react to the change. Because Prodigy is event driven, processes only use processor time when there is something new for them to do, meaning that processes that *are* running run quicker.

What Attributes Does A Signal Have?

Signals can have a great number of attributes. These range from the most basic, such as name and description, through such things as type (integer, real, digital, etc.) and source (serial I/O driver, Ethernet, OPC, etc.) to such things as recording strategies, smoothing options and scaling factors. All of these attributes can be created and modified by the Signal Database Configuration program.

Overview

The Signal database Configuration program is shown below, with the screen as it would look when no signals exist in the database and no drivers are loaded. The main functions of the program are accessed from the toolbar along the top of the screen.



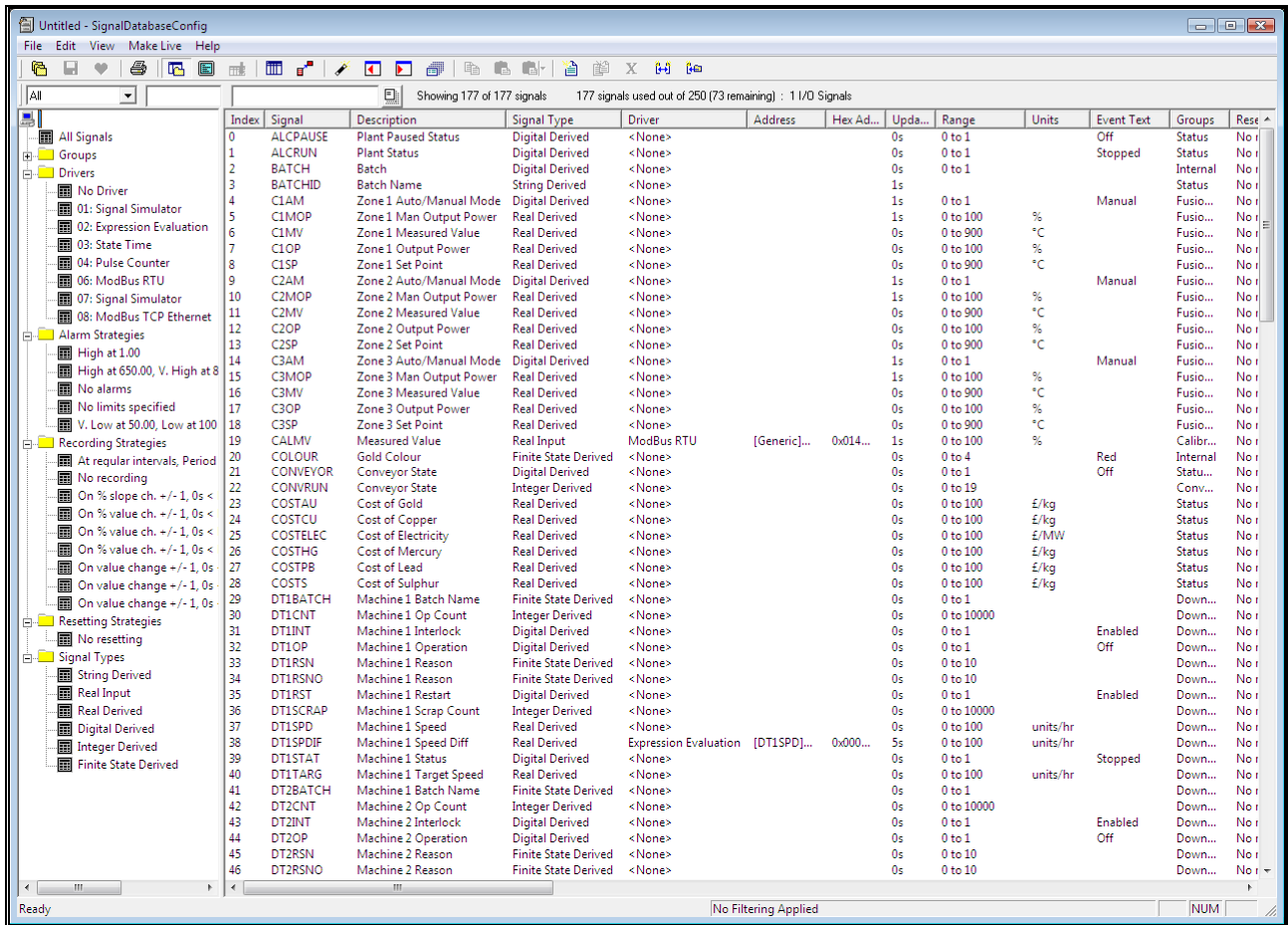
The area at the left hand side of the screen presents a variety of filtering options that vary according to what is defined in the signals (see later).

The main area of the screen is a spreadsheet type view of the signal database, showing the attributes for each signal in the database.

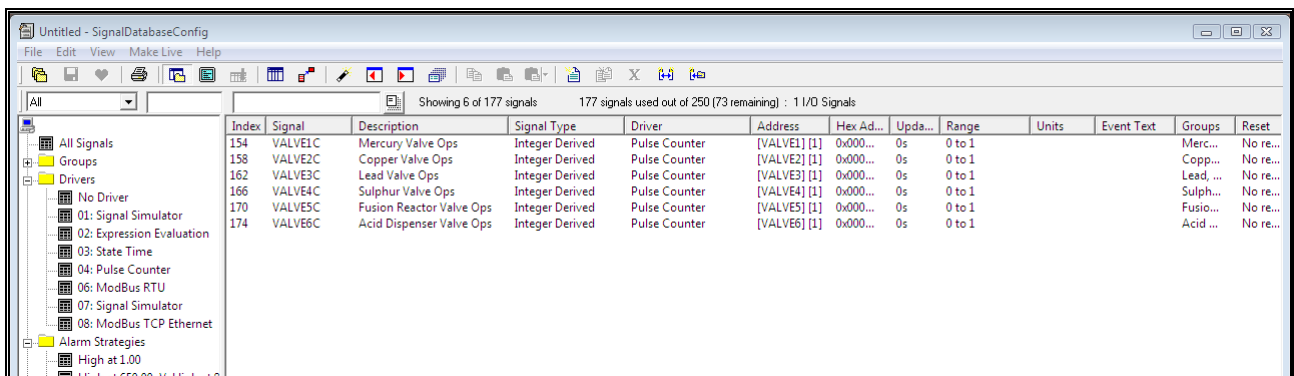
The attributes (columns) that are displayed can be selected by the user and saved as a user-defined “view”. The box below the toolbar that contains the word “All” is a dropdown list of all the pre-defined and user-defined views.

The two boxes to the right of the View selection list provide further filtering options that are based on signal name and description.

Below is a view with signals already in the signal database.




As you can see, the filtering lists in the left-hand pane are now filled with all of the different options that have been defined within the signals themselves. Each item in the lists is a unique set of attributes that has been applied to one or more signals. Clicking on one (or more) of those attributes in the left-hand lists will filter so that only those signals having those attributes will be displayed. This is shown below, where the signals have been filtered to be only those having the Pulse Counter driver.

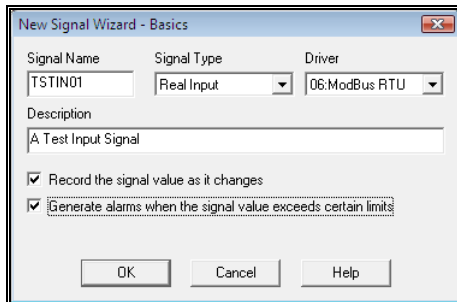


Creating New Signals

Creating Signals Manually

 To create a new signal click the “New” button. The **New Signal Wizard** (below) is then displayed. The following section explains how to create a new signal but the terminology used (what each field actually is, how to create new ranges, units, drivers etc.) is explained in more detail in a later section.

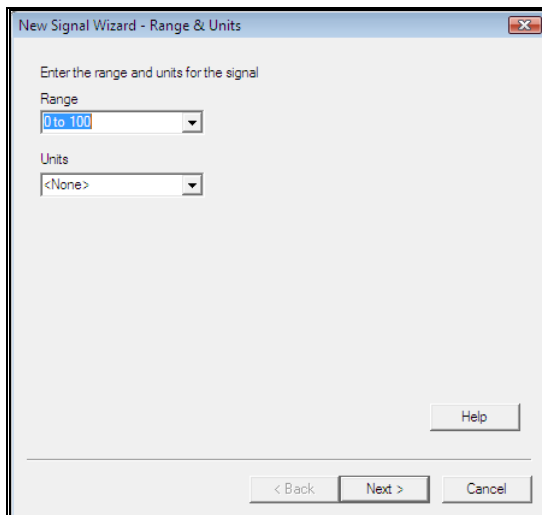
New Signal Wizard



To create **one** new signal, type its name into the **Signal Name** field (it’s possible to create many identical new signals at once, which will be explained later). Select the **Signal Type** and **Driver** and then enter a **Description** for the signal. If the driver that you want to use isn’t present in the list, select <New...> from the top of the list and install the required driver.

Check the relevant boxes if you want to attach a Recording Strategy and/or Alarm Strategy to the newly created signal(s). Finally press OK to proceed to the next part of the New Signal Wizard (below).

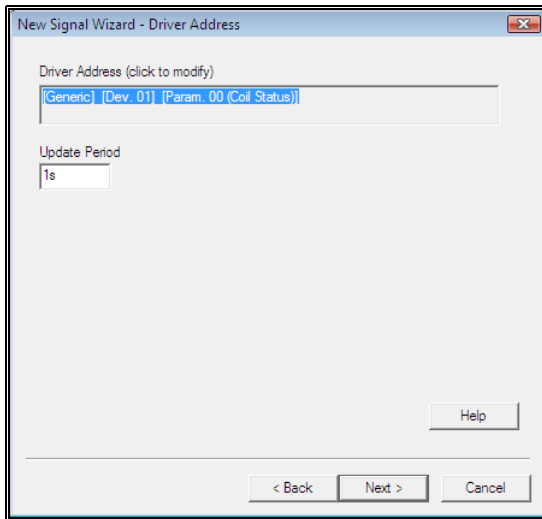
Range And Units



Select the range that is required for the signal. The drop-down list contains all ranges that have already been used within the signal database, along with a number of predefined standards. If you want to use a range that is not in this list click the <New...> option from the top of the list. This will show a dialog that allows you to enter the new range. Select the required units (or create new ones) in the same way.

Click **Next** to proceed to the next stage of the wizard (below).

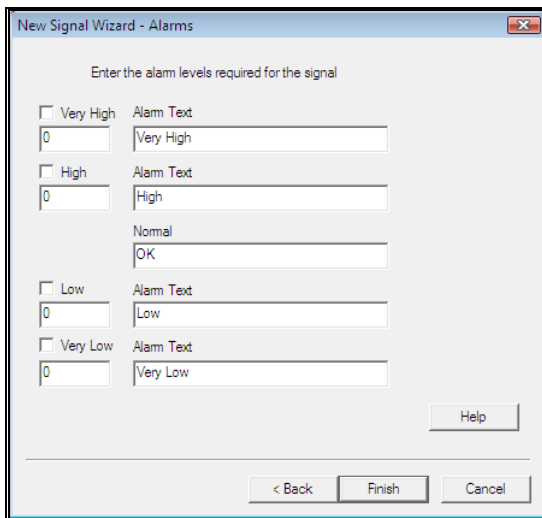
Driver Address



This section is not shown if no driver was selected on the first screen of the wizard.

Click in the **Driver Address** box to display the **Address Help** dialog for the driver that has been selected. Select the required parameters from that dialog and then enter the Update Period for the signal. Finally click **Next** to move on to the final part of the wizard (below).

Alarms



*This section is not shown if the option to assign an **Alarm Strategy** was not selected on the first screen of the wizard.*

This is the Alarms section. Check the boxes for the alarm limits that you want to set and enter values for those limits. Change the descriptive text for those limits too, if required. Finally press **Finish** to create the new signal(s) with all of the attributes that you have specified throughout the wizard.

You can click Back at any time to go back to previous stages of the wizard and make alterations or check what you have entered, although you cannot get right back to the initial screen.

Some of the screens may not look exactly like the ones shown here – for example if your signal is digital the range is fixed at 0 to 1 and you cannot alter it.

After Completion

If you had selected the option (on the first screen) to assign a recording strategy, a sensible strategy is automatically selected that varies according to the signal type. You can change this strategy (along with all other signal attributes) after the signal is created.

The newly created signal is shown below.

149	TIMEOPER	Waiting For Operator	Digital Derived	<None>		0s	0 to 1		Off
150	TIMETOT	Total Time Batch Running	Real Derived	State Time	[ALCRUN...]	0x000...	1s	0 to 3600	seconds
151	TIMEWAIT	Total Time Waiting	Real Derived	State Time	[TIMEOP...]	0x000...	1s	0 to 3600	seconds
152	TSTINO1	A Test Input Signal	Real Input	ModBus RTU	[Generic]...	0x000...	1s	0 to 100	
153	VALVE1	Mercury Valve	Digital Derived	<None>		0s	0 to 1		Closed
154	VALVE1AL	Valve 1 Maintenance	Digital Derived	Expression Evaluation	[VALVE1...]	0x000...	1s	0 to 1	Off
155	VALVE1C	Mercury Valve Ops	Integer Derived	Pulse Counter	[VALVE1] [1]	0x000...	0s	0 to 1	

Creating Many Signals At Once

To create a block of similar signals at once, instead of entering the exact name of the signal use the **&** character to represent the position at which an auto-incrementor will occur. Using one **&** will allow the creation of incrementors from 0 to 9, using two consecutive **&**'s (i.e. **&&**) will allow 00 to 99 and so on. An example is shown below.

This will cause the following dialog to be displayed after **OK** is pressed.

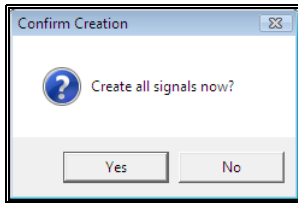
Enter the range of signals that you want to create, which in this example will create signals named TST01 to TST05. Note that if any of those signal names already exist they will simply be skipped. This means that if you know you need to increase a range of signals, you can simply enter the start and end numbers and any signals that exist will be untouched and any that don't exist will be created.

After clicking OK the wizard proceeds as before, except that at the end the whole range of signals will be created (below).

150	TIMETOT	Total Time Batch Running	Real Derived	State Time	[ALCRUN] [1] [No reset]	0x00000000	1s	0 to 3600	
151	TIMEWAIT	Total Time Waiting	Real Derived	State Time	[TIMEOPER] [1] [No reset]	0x00000000	1s	0 to 3600	
152	TST01	Test Signals	Real Input	ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Coil Status)]	0x00000000	1s	0 to 100	
153	TST02	Test Signals	Real Input	ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Coil Status)]	0x00000000	1s	0 to 100	
154	TST03	Test Signals	Real Input	ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Coil Status)]	0x00000000	1s	0 to 100	
155	TST04	Test Signals	Real Input	ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Coil Status)]	0x00000000	1s	0 to 100	
156	TST05	Test Signals	Real Input	ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Coil Status)]	0x00000000	1s	0 to 100	
157	VALVE1	Mercury Valve	Digital Derived	<None>		0s	0 to 1		
158	VALVE1AL	Valve 1 Maintenance	Digital Derived	Expression Evaluation	[VALVE1C] [\$VALVE1MC < \$VALVE1C]	0x00000000	1s	0 to 1	

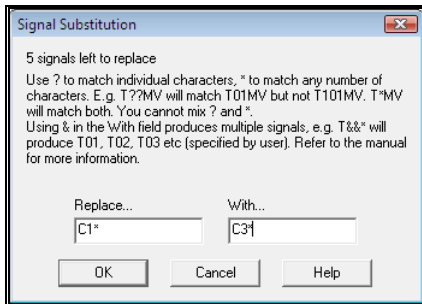
Creating Signals By Duplicating Existing Ones

To create The Signal database Configuration program is shown above, with the screen as it would look when no signals exist in the database and no drivers are loaded. The main functions of the program are accessed



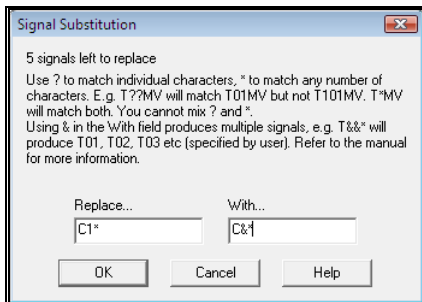
Substitution

The **Substitution** method is more flexible. In the example that we're using here, we can rename all of the C1... signals that we have selected to be C3... in one go. The dialog is as shown below.

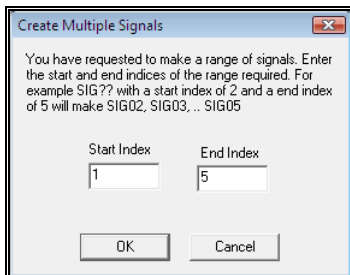


Filling the fields in as shown here will replace all occurrences of C1 with C3, no matter what follows the C1, so we will get C3AM, C3MOP, and so on.

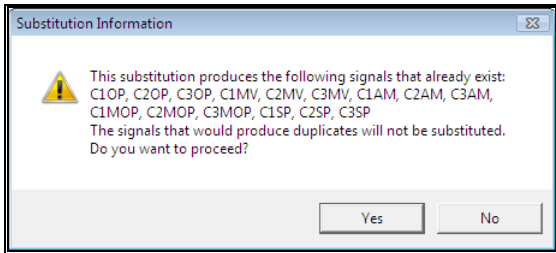
Wildcard substitution also allows you to create multiples of each signal. Suppose we enter values as shown below.



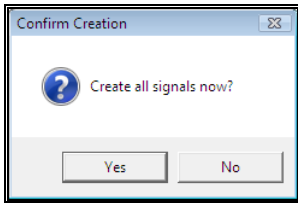
This will present the dialog that allows us to choose the number of signals that we want to create.



Entering the values shown here means we want to create C1AM, C2AM, C3AM, C4AM, C5AM, C1MOP, C2MOP, C3MOP, and so on. Looking back you will see that C1, C2 and C3 already exist for those signals, but that is no problem – they will simply be skipped. Using the values shown here will mean that we will end up with C1 to C5 for each of the signals selected. A warning dialog appears informing you of what is about to happen.



Finally the Confirm Creation dialog is shown to confirm that you want to create the signals.



The range of five signals that we initially selected will have been duplicated to create five copies of those signals named C1...C5


3	BATCHID	Batch Name	String Derived	<None>	1s		Status	No re..			
4	C1AM	Zone1 Auto/Manual Mode	Digital Derived	<None>	1s	0 to 1	Manual	Fusio...			
5	C1MOP	Zone1 Man Output Power	Real Derived	<None>	1s	0 to 100	%	Fusio...			
6	C1MV	Zone1 Measured Value	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
7	C1OP	Zone1 Output Power	Real Derived	<None>	0s	0 to 100	%	Fusio...			
8	C1SP	Zone1 Set Point	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
9	C2AM	Zone2 Auto/Manual Mode	Digital Derived	<None>	1s	0 to 1	Manual	Fusio...			
10	C2MOP	Zone2 Man Output Power	Real Derived	<None>	1s	0 to 100	%	Fusio...			
11	C2MV	Zone2 Measured Value	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
12	C2OP	Zone2 Output Power	Real Derived	<None>	0s	0 to 100	%	Fusio...			
13	C2SP	Zone2 Set Point	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
14	C3AM	Zone3 Auto/Manual Mode	Digital Derived	<None>	1s	0 to 1	Manual	Fusio...			
15	C3MOP	Zone3 Man Output Power	Real Derived	<None>	1s	0 to 100	%	Fusio...			
16	C3MV	Zone3 Measured Value	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
17	C3OP	Zone3 Output Power	Real Derived	<None>	0s	0 to 100	%	Fusio...			
18	C3SP	Zone3 Set Point	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
19	C4AM	Zone1 Auto/Manual Mode	Digital Derived	<None>	1s	0 to 1	Manual	Fusio...			
20	C4MOP	Zone1 Man Output Power	Real Derived	<None>	1s	0 to 100	%	Fusio...			
21	C4MV	Zone1 Measured Value	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
22	C4OP	Zone1 Output Power	Real Derived	<None>	0s	0 to 100	%	Fusio...			
23	C4SP	Zone1 Set Point	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
24	C5AM	Zone1 Auto/Manual Mode	Digital Derived	<None>	1s	0 to 1	Manual	Fusio...			
25	C5MOP	Zone1 Man Output Power	Real Derived	<None>	1s	0 to 100	%	Fusio...			
26	C5MV	Zone1 Measured Value	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
27	C5OP	Zone1 Output Power	Real Derived	<None>	0s	0 to 100	%	Fusio...			
28	C5SP	Zone1 Set Point	Real Derived	<None>	0s	0 to 900	°C	Fusio...			
29	CALMV	Measured Value	Real Input	ModBus RTU	[Generi]..	0x014...	1s	0 to 100	%	Calibr...	No re..

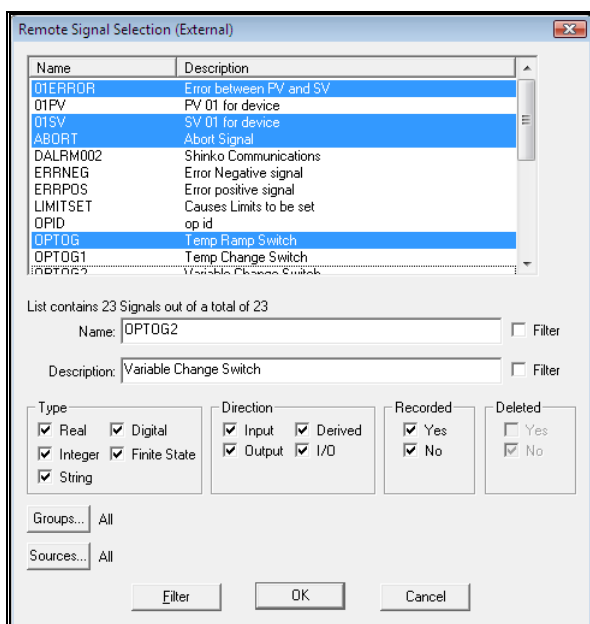
Examples Of Substitution Strings

Existing signal	à Replace(left field)	à With(right field)	à Produces
01MV	à ??*	à &&*	à 02MV, 03MV etc.
01MV	à *MV	à &&*	à 02MV, 03MV etc.
01MV, 01SP	à 01*	à &&*	à 02MV, 03MV, 02SP, 03SP etc.
MV01	à MV??	à MV&&	à MV02, MV03, etc.
MV01, SP01	à *01	à *&&	à MV02, MV03, SP02, SP03 etc.

Importing Signals From Other Systems

You may be configuring a system that contains elements that you have configured before on other jobs, in this instance we're thinking mainly about signals. If you have already configured blocks of signals that serve the same purpose on other jobs it's tiresome to have to enter them again, one by one. Signal Database Configuration allows you to import such signals directly from the saved signal database file (signals.ass) for that job into your current job.

To do this simply click the **Import Signals** button . This will bring up a browser window that will allow you to browse for and select a signals.ass file. On selection of the required signals.ass file the program will load the signals from it into a SigScan dialog. Use this to select the signals that you want to import (in this mode SigScan allows multiple selection of signals).

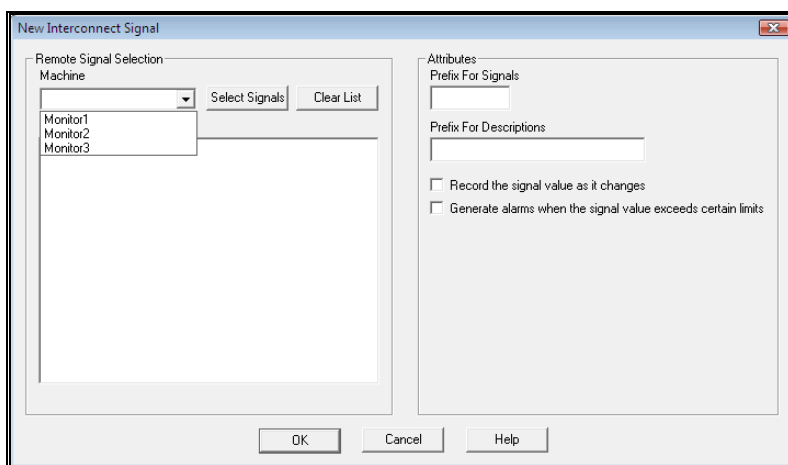


Adding Interconnect Signals

Prodigy **Interconnect** is a facility that allows two or more Prodigy systems to share signal data between themselves across a network connection. For example, there may be three different control & monitoring systems all linked to a central supervisory system. All of the actual external I/O takes place on the control & monitoring machines but some (or all) of the signal data from these machines are transmitted to (and/or from) the supervisory machine. Each of the control systems may have signals called AI01, AI02 and so on. On the supervisory machine these signals must be mapped to signals of different names (a signal name can only occur once in a given system), for instance M1AI01, M2AI01, M3AI01.

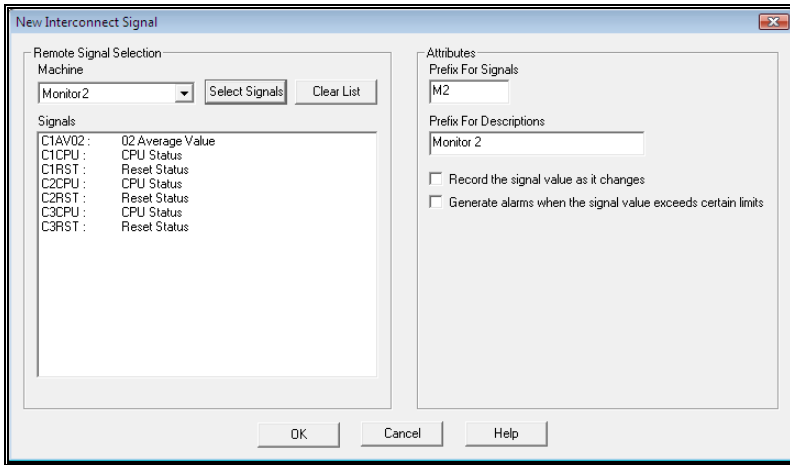
To add Interconnect signals the first thing that you must do is to actually have an interconnect system configured. This is because adding Interconnect signals requires that the signals already exists on the other machines, and access to them is via the Network driver that is used to transmit the signal data.

To add Interconnect signals click the **New Interconnect** button . The dialog shown below will appear.



The drop down list in the top left corner contains all machines to which an interconnect is established. Select the machine from which you want to add signals and the program will attempt to load in the signal database

from that machine. If it succeeds a SigScan dialog will be displayed and you can select (multiple) signals from SigScan to add into your signal database. The signals that you have selected are shown in the dialog (below) – you can add to this list by clicking **Select Signals** and choosing more signals or, if you've made a mistake, you can click **Clear List** to start again. If you change Machine it will clear your list and display a new SigScan from the newly selected machine (you can only select signals from one machine at a time).



You can (optionally) enter prefix values in the area at the right of the screen. Take care with the signal prefix, as the total length of a signal name can only be eight characters. If any of your selected signals are already eight characters, and you really need to prefix them, then you're in trouble!

Check the boxes if you want to have an alarm and/or recording strategy with the signals.


When you click OK the signals will be created. They will have the same name as their source signal but prefixed with the optional prefix text that you entered, and similarly for the descriptions. If you have checked the recording/alarm strategy boxes the signals will be given the same strategies as their respective source signals, all other relevant attributes (units, range, event text and so on) are taken from the source signal too.

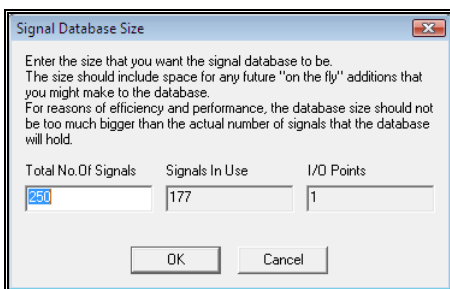
The created signals are shown below.

90	M2C1AV02	Monitor 2: O2 Average Value	Real Input	Network 2	MONITOR2 : C1AV02	0x00...	5s	vol %	0 to 20	Valve
91	M2C1CPU	Monitor 2: CPU Status	Digital Input	Network 2	MONITOR2 : C1CPU	0x00...	15s	Off-Line	0 to 1	Valve
92	M2C1RST	Monitor 2: Reset Status	Digital Output	Network 2	MONITOR2 : C1RST	0x00...	0s	Off	0 to 1	Valve
93	M2C2CPU	Monitor 2: CPU Status	Digital Input	Network 2	MONITOR2 : C2CPU	0x00...	15s	Off-Line	0 to 1	Stirrer
94	M2C2RST	Monitor 2: Reset Status	Digital Output	Network 2	MONITOR2 : C2RST	0x00...	0s	Off	0 to 1	Stirrer
95	M2C3CPU	Monitor 2: CPU Status	Digital Input	Network 2	MONITOR2 : C3CPU	0x00...	15s	Off-Line	0 to 1	Pump
96	M2C3RST	Monitor 2: Reset Status	Digital Output	Network 2	MONITOR2 : C3RST	0x00...	0s	Off	0 to 1	Pump

The Address field shows the computer name and source signal name, for example MONITOR2: C1CPU.

Increasing The Database Size

The Signal Database has a defined upper limit as to the number of signals that it can hold. This upper limit can be set so that the database has some spare capacity. For instance, the database size can be set to 200, while the database itself contains only 150 signals. This leaves room to add 50 more signals later without having to increase the database size. To add extra signals into the live database (see later) it is essential to have some spare capacity, but do not create too much as extra capacity means extra unused memory. To change the size of the database click the Database Size button . This will display the following dialog.



The dialog spells out the number of signals the database can hold, the number of signals that it does hold and the number of signals that are I/O points (this latter is for reference only). You can increase the database size

to anything you like by entering a value into the **Total No. Of Signals** box. *Remember that the database should not have **too** much spare capacity!*

You could also reduce the database size here too, provided that it is not reduced to a value **lower** than the **Signals In Use** value (it won't let you).

After you click OK the program recreates its dataspace with the value you have entered. Status displays inform you of its progress.

Changing The Attributes Of A Signal

Data Entry Basics

When you click on a field within a row one of two things happens. If the row you click in is not already the highlighted row, it will become the highlighted row. If the row you click in *is* the highlighted row, the cell you click in goes into the editing state.

The method of data entry within the cells on the display varies according to the type of data that the cell represents. With free-format data, such as the description, it is simply a case of typing within the cell (below).

Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Range	Units	Event Text	Groups	Reset
154	VALVE1C	Mercury Valve Ops	Integer Derived	Pulse Counter	[VALVE1] [1]	0x000...	0s	0 to 1			Merc...	No re...
158	VALVE2C	Copper Valve Ops	Integer Derived	Pulse Counter	[VALVE2] [1]	0x000...	0s	0 to 1			Copp...	No re...
162	VALVE3C	Lead Valve Ops	Integer Derived	Pulse Counter	[VALVE3] [1]	0x000...	0s	0 to 1			Lead, ...	No re...
166	VALVE4C	Sulphur Valve Ops	Integer Derived	Pulse Counter	[VALVE4] [1]	0x000...	0s	0 to 1			Sulph...	No re...
170	VALVE5C	Fusion Reactor Valve Ops	Integer Derived	Pulse Counter	[VALVE5] [1]	0x000...	0s	0 to 1			Fusio...	No re...
174	VALVE6C	Acid Dispenser Valve Ops	Integer Derived	Pulse Counter	[VALVE6] [1]	0x000...	0s	0 to 1			Acid ...	No re...

Other data, such as the signal type, is take from a list of pre-determined values that drops down when you click in the field (below).

Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Range	Units	Event Text	Groups	Reset
154	VALVE1C	Mercury Valve Ops	Integer Derived	Pulse Counter	[VALVE1] [1]	0x000...	0s	0 to 1			Merc...	No re...
158	VALVE2C	Copper Valve Ops	Integer Derived	Counter	[VALVE2] [1]	0x000...	0s	0 to 1			Copp...	No re...
162	VALVE3C	Lead Valve Ops	Finite State Derived	Counter	[VALVE3] [1]	0x000...	0s	0 to 1			Lead, ...	No re...
166	VALVE4C	Sulphur Valve Ops	Finite State I/O	Counter	[VALVE4] [1]	0x000...	0s	0 to 1			Sulph...	No re...
170	VALVE5C	Fusion Reactor Valve Ops	Finite State Input	Counter	[VALVE5] [1]	0x000...	0s	0 to 1			Fusio...	No re...
174	VALVE6C	Acid Dispenser Valve Ops	Finite State Output	Counter	[VALVE6] [1]	0x000...	0s	0 to 1			Acid ...	No re...

Some data consists of a complex set of interrelated options and is therefore entered via dialogs. An example of this is the signal address field (below).

Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Range	Units	Event Text	Groups	Reset
154	VALVE1C	Mercury Valve Ops	Integer Derived	Pulse Counter	[VALVE1] [1]	0x000...	0s	0 to 1			Merc...	No re...
158	VALVE2C	Copper Valve Ops	Integer Derived	Pulse Counter	[VALVE2] [1]	0x000...	0s	0 to 1			Copp...	No re...
162	VALVE3C	Lead Valve Ops	Integer Derived	Pulse Counter	[VALVE3] [1]	0x000...	0s	0 to 1			Lead, ...	No re...
166	VALVE4C	Sulphur Valve Ops	Integer Derived	Pulse Counter	[VALVE4] [1]	0x000...	0s	0 to 1			Sulph...	No re...
170	VALVE5C	Fusion Reactor Valve Ops	Integer Derived	Pulse Counter	[VALVE5] [1]	0x000...	0s	0 to 1			Fusio...	No re...
174	VALVE6C	Acid Dispenser Valve Ops	Integer Derived	Pulse Counter	[VALVE6] [1]	0x000...	0s	0 to 1			Acid ...	No re...

Pulse Counter Details

Source signal:

Count pulse when value changes to:

0 1 Either

With free-format and list-based entry, the data is not accepted until you move onto a different field. If you decide you don't want the change you have just made (and you haven't moved to a different field) you can press Esc to undo the change.

You can navigate around the fields in the grid by using the tab and enter (or return) keys. Pressing tab moves from left to right on the same row, while pressing shift+tab moves from right to left. When you reach the end of the row, at either the left hand or right hand side, tabbing will stop. It does not move onto the next (or previous) line.

Pressing enter moves vertically downwards within the same column, while pressing shift+enter moves vertically upwards. When the cursor moves into a new field the field is placed into its relevant edit mode, which may involve bringing up a dialog. You cannot move to the next field without first exiting that dialog. As with tabbing, reaching the top or bottom of a column will not cause an automatic jump into the next (or previous) column.

Field By Field In The Signal Database

This section will describe each field that is available in Signal Database Configuration, along with its method of data editing. The fields are presented in the order that they initially appear in the predefined column set called **All**.

Index

This is the signal's internal index. The user does not usually need to know this, but it is presented for reference. The signal database is stored internally with signals sorted alphabetically by name. The Index is the signal's position within this sorted list. You cannot alter the signal's index, so no editing of this field is possible. A given signal's index may change though, for example if a new signal is created which comes alphabetically sooner in the list.

Signal

This is the signal name. Once created a signal cannot be renamed. To effect a rename you must create a new signal with the name that you want and then do a Copy Attributes from the existing to the new signal. Finally, delete the old signal. Be careful about anything else that may have referenced the old signal (e.g. **Copy To** or some drivers such as **Expression Evaluation**).

Description

This is free format text that describes the signal and has to be less than 32 characters long. To change the description click in the cell and enter the new text.

Signal	Description
VALVE6C	Acid Dispenser Valve Ops

Signal Type

This is the type *and* direction of the signal. Types can be

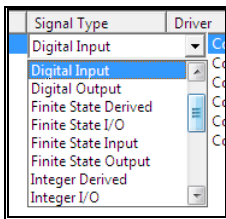
- **Digital.**
The signal has only the values 0 or 1 and is generally used to control or monitor things that have only two states (generally Running/Stopped, On/Off, Open/Closed and so on). For each of the states an item of text can be defined that describes that state (e.g. "Off" and "On"). No units are associated with this type.
- **Finite State.**
Similar to Digital but the signal has a predefined number of discrete states (e.g. Off/Slow/Medium/Fast). For each state an item of text can be defined that describes that state (e.g. again, "Off", "Slow", "Medium" and "Fast"). The maximum number of states is 100 and no units are associated with this type.
- **Integer.**
This is an actual numeric value of integer type (i.e. it can have no fractional parts). The range is unlimited and units can be associated with it, if required.
- **Real.**
The same as Integer but the value is not limited to integers, instead it is a double precision floating point value.
- **String.**
String variables have no range and no units, and instead of numeric values they hold items of text. The maximum length of an item of text is 80 characters.

Direction can be

- **Input.**
Values are read in from external hardware via a driver within the Prodigy system. These drivers typically use serial communications but other options include Ethernet comms, plug-in cards and OPC. Input signals must have an **Update Rate** of greater than 0 otherwise they will not be polled by the driver that is associated with it.

- **Output.**
Values are written out to external hardware via the same driver that handles the inputs. Values are written out as and when changes occur within Prodigy and in addition they can be written out at a regular interval specified by the Update Rate.
- **I/O.**
Values are read *and* written to external hardware. They are read in at the rate specified by Update Rate and written out only when they are changed within Prodigy.
- **Derived.**
The signal is not associated with any external hardware. The signal value is used only within Prodigy and may be set and/or read by any one of a number of Derived Drivers, SLANG, Display Builder buttons, Batch Data Recorder and so on.

The signal type (& direction) is selected from a drop down list.

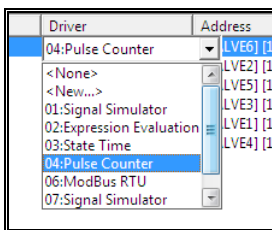


Driver

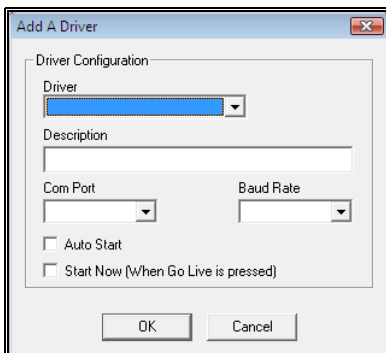
This field specifies the driver that is to be associated with the signal. If the signal is to communicate with external hardware (it should therefore be of type **Input**, **Output** or **I/O**) then the driver appropriate for that hardware should be selected. If the signal is to be manipulated by one of Prodigy's Derived Drivers (and is of type **Derived**) then again the relevant driver should be selected. If the signal is only to be manipulated by other parts of Prodigy (SLANG, BDR, etc) then it does not need to have a driver.

Drivers are selected from a dropdown list that presents all of the drivers that are currently installed for the current configuration. In addition, an entry in the list (the first entry) gives the ability to add a new driver into the system (as an alternative to this, drivers can be loaded via the Driver Configuration program). Each driver can handle one communications loop and where multiple devices can be hung off the same loop (usually referred to as multi-drop) the driver can handle this too. See the **Driver Configuration** section of the manual for more detailed information.

Below is the dropdown list of drivers.



If you select **<New...>** the dialog for adding a new driver is displayed (below).



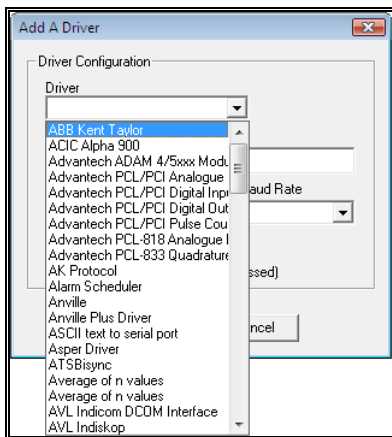
The dropdown list of **Drivers** shows every driver currently available within Prodigy. Select the driver that you require and then give it a description. Set the **Com Port** and **Baud Rate** parameters to appropriate values and check the boxes for **Auto Start** and/or **Start Now** (sensible defaults are entered by Prodigy for the Baud Rate and Description fields, though your description should also indicate what this loop is used for).

The Auto Start box tells the driver that it is to start automatically each time Prodigy is started. This is usually the case once the system is live, but may not be required during design time when all of the hardware may not be available.

The **Start Now** box tells Prodigy to start the driver immediately that the Go Live button is pressed in Signal Database Configuration.

Press **OK** when you have completed this dialog.

Below is the dialog showing part of the list of drivers that is available.



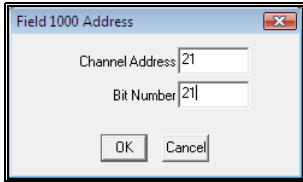
Address

The next field is **Driver Address**, abbreviated to just **Address**. For serial I/O drivers this describes which of the available parameters within the attached hardware that the signal is to reference. For derived drivers it describes the various parameters used to perform the derived functionality. Below is an example of the Driver Address column, showing a variety of signals with varying drivers.

Driver	Address
Expression Evaluation	[LVLABATH] [SLVLABATH >= 800]
Expression Evaluation	[LVLABATH] [SLVLABATH <= 100]
Pulse Counter	[VALVE6] [1]
Expression Evaluation	[VALVE6] [SLVLADISP > 0 AND SVALVE6]
Pulse Counter	[VALVE2] [1]
Expression Evaluation	[STAGE] [1]
Expression Evaluation	[FILTTIM] [SFILTMP < SFILTTIM]
State Time	[FILTER] [1] [No reset]
Expression Evaluation	[BATC] [(SFLASH1 + 1) * (SFLASH1 < 9)]
Expression Evaluation	[PUMP3] [SLVLABATH > 0 AND SPUMP3 = 1]
Pulse Counter	[VALVE5] [1]
State Time	[FUSRUN] [1] [No reset]
Signal Simulator	
Signal Simulator	
Pulse Counter	[VALVE3] [1]
Expression Evaluation	[DT1SPD] [SDT1SPD - SDT1TARG]
Expression Evaluation	[DT2SPD] [SDT2SPD - SDT2TARG]
Expression Evaluation	[DT3SPD] [SDT3SPD - SDT3TARG]
ModBus RTU	[Generic] [Dev. 01] [Param. 00 (Input Register)]

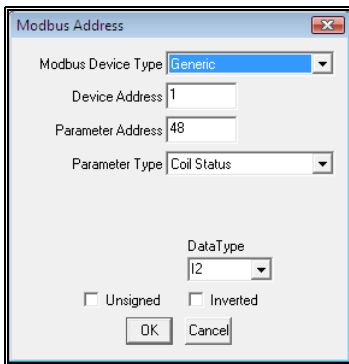
The information displayed for Driver Address varies from driver to driver, but is always the most pertinent information for that driver. We won't list here what the information is for each type of driver, it should become apparent to the user what the information represents as they use each type of driver address dialog.

Clicking in the Address field displays a dialog that is relevant to the type of driver that has been selected. The Driver Address dialogs can be very different between different types of driver. The addressing for the Field 1000 driver is very simple and is shown below.

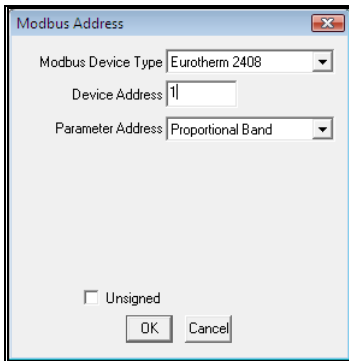


For the Modbus RTU driver (and others) there is the choice of addressing either generically or by specific device type. The specific devices are more user-friendly but generic addressing allows you to access any modbus-protocol device, even if it is not explicitly handled by Prodigy. The two forms of the Modbus RTU dialog are shown below, the fields within the dialog changing according to what is selected in the dropdown list at the top of the dialog.

Generic...



And Eurotherm specific...



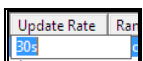
Hex Address

The hex address is provided for reference only and can be useful when developing new drivers. It can also sometimes be useful to sort the columns based on the hex address, certainly within a given driver.

Update Rate

The Update Rate has been mentioned earlier. It specifies how often the driver reads or writes values to the hardware, or, in the case of Derived drivers, how often the derived calculation is performed. The update rate is entered as a time period value, which in Prodigy is a real number followed by the letter h, m or s (for hours, minutes and seconds). For example 30 seconds is 30s (or 0.5m if you like, though it will redisplay as 30s). 90 seconds can be written as 90s, 1.5m or 1m30s – it will always redisplay as 1m30s.

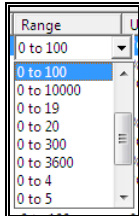
If an input or I/O signal has an Update Rate of 0s it will never be updated. A derived signal with an update rate of 0s may be updated if it has a **trigger signal** (Expression Evaluation has such a signal) but otherwise it will not be.



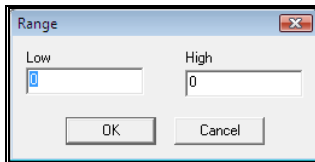
Range

A range can be applied to all signal types except digital and string. Digitals always have a range of 0 to 1 and strings have no range at all. Finite states can have a range that runs from any value to any value so long as the actual span of the range is no greater than 100 (e.g. 0 to 99). For other signal types the range is used only for labelling axes on trends etc. it will not limit the data that is taken by a system and is not used to perform any real scaling (for that see Linear Interpolation, later).

Select the required range from the dropdown list of ranges that are already in use, or create a new one by selecting <New...>. The dropdown list is shown below...

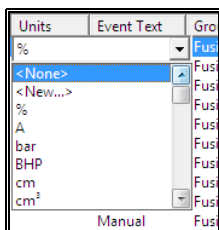


and the New Range dialog...

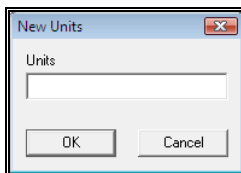


Units

Units may be applied to all signals that are of type Integer or Real. The units are just descriptive text and do not imply that any kind of conversion takes place. Units are selected from a dropdown list that contains both units that already exist within the configuration and standard predefined units. As well as this there is an option to create a new Units flag or to have none. The list is shown below.



The dialog for entering new units is very simple and is shown below.



Event Text

Event text is applied to signals of type Digital and Finite State. As described elsewhere, this text is used to describe the actual states that the signal can take (there are two states in a digital and up to 100 in a finite state). As such the range of the signal should be defined before the event text is defined. The dropdown list will show the event texts that are already being used, in blocks that match the span of the range (below).

Range	Units	Event Text	Groups	Re
0 to 900	°C		Fusio...	No
0 to 100	%		Calibr...	No
0 to 4		Red	Internal	No
0 to 1		Off, On		No
0 to 19		No Recipe,		No
0 to 100	£/kg		No
0 to 100	£/kg	Off, On		No
0 to 100	£/MW		No
0 to 100	£/kg	Off, Paused		No
0 to 100	£/kg		No
0 to 100	£/kg	Stopped, Running		No
0 to 1			No

Range	Units	Event Text	Groups	Re
0 to 4		Red		No
0 to 1		<New...>		No
0 to 19			No
0 to 100	£/kg	Red		No
0 to 100	£/kg	Yellow		No
0 to 100	£/MW	Green		No
0 to 100	£/kg	Blue		No
0 to 100	£/kg	Violet		No
0 to 100	£/kg		No

If the event text that you want isn't already used (i.e. it isn't in the list) select <New...> to create a new one (below).

Event Text

Event Texts

- [00] Red
- [01] Yellow
- [02] Green
- [03] Blue
- [04] Violet

New Text

The list shows the event text entries already in use for that signal. Click on a row and type new text into the **New Text** box. Click on each row to provide new text for them all and then press **OK** when finished.

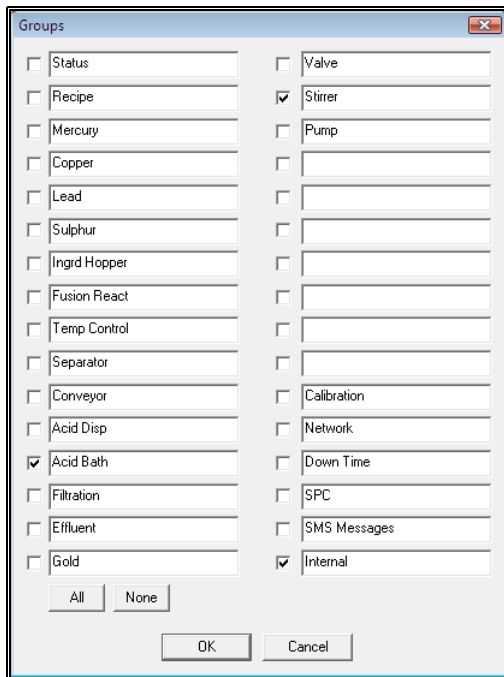
Groups

Groups provide a mechanism for identifying signals that perform a particular function, pertain to a certain area of plant and so on. There are 32 definable groups and a signal can belong to none, any or all of these groups. Groups are used only for filtering during selection via SigScan or here in Signal Database Configuration. In the example below you can see groups for **Acid Bath**, **Stirrer**, **Internal**, **Acid Disp**, **Valve** and so on. Some signals belong to one of those groups, some to two (e.g. **Acid Bath & Stirrer**) some to three of them (**Acid Bath**, **Stirrer** and **Internal**) and so on.

- Groups

 - Acid Bath
 - Acid Bath, Stirrer
 - Acid Bath, Stirrer
 - Acid Bath, Stirrer
 - Acid Bath, Stirrer
 - Acid Bath, Stirrer, Internal
 - Acid Disp
 - Acid Disp, Internal
 - Acid Disp, Internal
 - Acid Disp, Valve
 - Acid Disp, Valve
 - Acid Disp, Valve
 - Calibration
 - Conveyor, Internal
 - Copper
 - Copper, Valve
 - Copper, Valve
 - Copper, Valve
 - Copper, Valve
 - Down Time
 - Down Time
 - Down Time

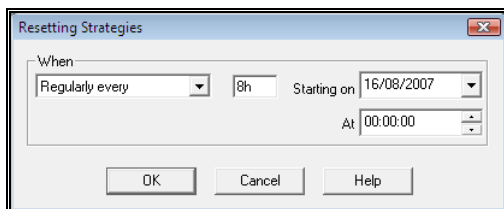
Groups are both selected and named via the same dialog, shown below (the current signal is in groups **Acid Bath**, **Stirrer** and **Internal**).



Reset

Resetting strategies allow signals to be reset at given intervals. This particularly applied to counters and timers which may be reset at midnight, for instance, to provide daily totals and elapsed times. In systems that use Batch Data Recorder this usually performs signal resetting when a batch starts or ends. Another alternative is that signals can be reset by SLANG when required. In systems that do not use BDR or SLANG the only way to reset a signal is by use of a resetting strategy.

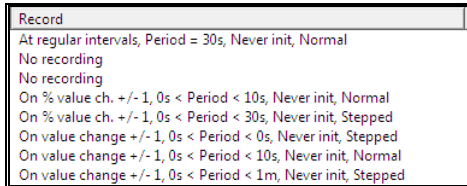
The dialog for entering resetting strategies is shown below.



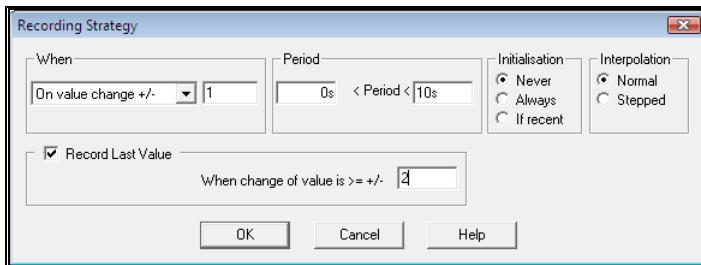
In this example the signal is reset every 8 hours from midnight, giving a sort of three-shift resetting strategy.

Record

Signal values may need to be recorded to disk file so that a permanent archive of what occurred can be maintained, either for analysis or, for example, compliance with regulations (e.g. food and pharmaceutical industry regulations). Prodigy allows you to create a recording strategy that is appropriate to the type of signal and both the frequency and amount by which it changes. In its simplest form Prodigy can record a signal's value each and every time it changes. For a signal that fluctuates rapidly but insignificantly this would create too many recordings, so a better idea is to make a recording only when the signal changes by a significant amount. This can be specified in a number of ways, including a change by a certain percentage or a change by a certain fixed amount. In addition, a regular recording can be forced, so that even if the signal's value doesn't change (significantly) for a while, a recording is still made every now and then. An example of some available recording strategies is shown below.



Below is the dialog used for creating recording strategies.



The fields are as follows.

- **When.**

This determines under what circumstances a recording is made and includes things such as value change, % value change, rate of change and so on. The box to the right of the dropdown list is where you should enter the value against which the change is compared.
- **Period.**

This allows you to limit how frequently a recording is made. The first box indicates that a recording will not be made if it is within this time of the last recording having been made. For instance, if it were 1s (one second) then recordings would be made no more frequently than once a second, no matter how frequently the signal was changing. It is usual to leave this value at 0s, meaning that every recording is made (provided it falls within the significance specified in **When**, above). The second box indicates that a recording should be made at least this often. In the example shown, a recording will be made every 10 seconds, even if the signal is not changing.
- **Initialisation.**

This allows you, on system startup, to reset a signal to the value that it had at system shutdown. This is useful for instance with counters, where the count should continue after the system is restarted. The options are **Never**, where initialisation will never occur; **Always**, where initialisation always occurs; and **If Recent**, where initialisation will occur if it is less than 1 hour since shutdown. **If Recent** is rarely used.
- **Interpolation.**

Suppose a signal is an analogue value representing temperature. A recording will be made at one temperature, and later (determined by the recording strategy) another recording will be made at a different temperature. It is fair to assume that the temperature changed steadily between the two values and that they can be represented on a graph by a straight line joining the two points. Interpolation could then be used to calculate the temperature at some time between the two recordings. This is referred to as **Normal** interpolation.

If, though, the signal represents not a measured value but a setpoint, the situation is different. The temperature being demanded at one moment may be 30° and later 100°, but the change in demand would be instantaneous, requiring that the graph be drawn as a step change. It would be drawn as a

flat line at 30° up until the precise moment that the change to 100° occurred, at which point it would be drawn as a vertical line from 30 to 100. This is known as **Stepped** interpolation and is always applied to **Digital, Finite State** and **Integer** signals, regardless of what is selected here.

- **Record Last Value.**
Suppose an analogue value hasn't changed (significantly) for some time, say 1 hour, and then suddenly rises. Under a **Normal** recording strategy the graph of that signal would show a gently sloping line from the last recorded value to the new, spanning a time of 1 hour. That is not actually what happened, so that representation is invalid. What we want to see is a flat line from one hour ago until the time at which the temperature began to rise, and this is what can be achieved by using an appropriate value in the **Record Last Value** field. The value should be greater than that specified in the **When** field, or else you may as well have just chosen a **Stepped** strategy.
Note that the problem as described here could also be avoided, or at least minimised, by having some degree of regular recording, but that could lead to too many recordings being made of a signal that rarely changes. It is all a matter of common sense and application.

Alarm

Sometimes when signal values exceed certain limits it is required that some sort of warning is given. In Prodigy this is governed by a signal's **Alarm Strategy**. Signals can be given four levels of alarm, **Very Low, Low, High** and **Very High** and for each of these states the value which triggers the alarm can be specified, along with text that describes the condition. The dialog for entering Alarm Strategies is shown below.

The fields are as follows.

- **Use.**
If this box is checked the signal will have an Alarm Strategy, otherwise it will not.
- **Very High, High, Low, Very Low.**
Check the boxes for the alarm levels that are to be used and enter values into the boxes below the check-boxes that specify the value of the alarm level.
- **Alarm Text.**
This is a description for the alarm level and this text is used when the alarm is displayed on the **Alarm Banner**. Suitable descriptions may simply be “High” and “Very High”, as shown, but could also be things such as “Full”, “Critical”, “Too Cold” and so on, depending on the application.
- **Annunciation.**
Annunciation (if required) can take the form of a flashing light, siren, or whatever is appropriate. In order to implement annunciation another signal must be associated with the alarm (or with a particular alarm state) of the signal in question. This annunciation signal is set to 1 whenever the source signal goes into a (relevant) alarm state. The annunciation signal should (usually) be of type Digital Output and be connected to the relevant piece of hardware (siren etc.) via an appropriate driver.

Select the annunciation signal(s) by clicking the **Select...** button. If you check the **All** box then all of the alarm states for this signal will be attached to one annunciation signal (selected against the **Very High** state, regardless of whether or not the Very High state is being used).

Priority.

Normally the Prodigy Alarm Banner shows the oldest unaccepted alarm, with all other alarms that are also unaccepted being “hidden”, their presence being indicated by a **+1 more**, **+2 more**, etc. on the alarm banner.

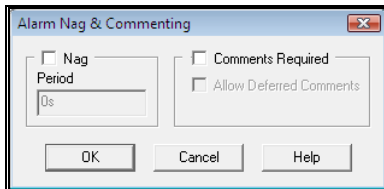


- This functionality can be overridden by use of **Alarm Priority**. If a priority is assigned to an alarm then alarms of a higher priority (larger number, 0 is lowest) appear in preference to alarms of a lower priority. So if a priority 0 alarm is already displaying and a priority 1 alarm becomes active, the new alarm will be displayed on the alarm banner even though it is newer than the existing alarm. The rule is that alarms are sorted by priority first and then by date (oldest comes first) within alarms of equal priority.
- **Accept On Revert**. Prodigy can be configured so that an alarm *automatically* accepts itself if it reverts (i.e. drops back out of alarm). To do this simply check the **Accept On Revert** box next to the relevant alarm state, or the **Accept All On Revert** box.

At the bottom of the dialog are three buttons that form the **Advanced** properties of Alarm Strategies. Alongside each button is some text that briefly describes what (if any) attributes have been configured within the relevant dialog. The buttons are as follows.

Nag & Comments

Alarm nag is the facility whereby if an alarm is accepted, but the signal remains in alarm, the alarm will become active again. If the alarm were attached to an annunciator, that would sound again. It is a mechanism by which operators can be reminded that an alarm condition still exists, even though it has been accepted.



To set up a Alarm Nag check the Nag box and enter a time period for the nag.

Alarm Commenting is a facility whereby operators can be required to enter a comment along with an alarm acceptance. The comment should take the form of why the alarm condition occurred, or what was done to fix it, or what other conditions existed within the plant and so on. It is a matter of protocol as to what the operator should enter here.

To enable Alarm Commenting check the **Comments Required** box. What then happens is that when the operator accepts an alarm, a dialog is displayed requesting a comment be entered. If the **Allow Deferred Comments** box has been checked, the operator is able to dismiss the request without entering anything. They can then go back later and enter a comment for the alarm by using the **Alarm History Viewer** program.

Deviation & Hold Offs

It is not always desirable for an alarm to be flagged the moment an alarm condition is entered. Sometimes the alarm is only really relevant if the state has been active for a period of time. For instance, it may be OK for a temperature to momentarily rise above a specific value, but if it remains that way for more than a few seconds it may start to cause problems. Specifying that an alarm should not sound until after the condition has been active for a while is known as **Alarm Hold Off**.

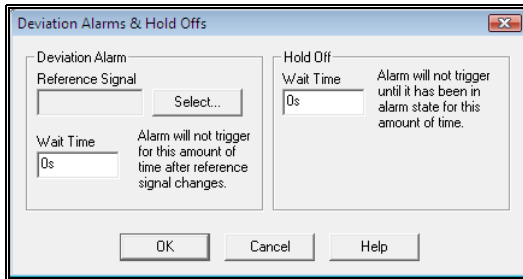
Other instances may require that an alarm becomes active not simply on a given value being reached, but on the difference between *two* values reaching a certain level. The most obvious example of this is the difference between a setpoint and a measured value. The alarm should not become active at any given value of the

measured value, but rather when the measured value varies by a given amount from the setpoint. This is known as alarm deviation.

Within alarm deviation there is also the notion of **Wait Time**. This is not the same hold off as previously described, but is the time that a measured value is given to attain the setpoint. Suppose a measured value is matching its setpoint at value A and then the setpoint is raised to value B. This does not necessarily mean that an alarm should be raised immediately. If the controller (for example) fails to get the measured value up to the setpoint's new value within a certain period of time, then *that* is what triggers the alarm.

If however the setpoint remains steady and the measured value drifts away from it, the alarm will trigger immediately. Deviation hold off is only appropriate when the setpoint moves.

The dialog for configuring these is shown below.



The alarm strategy should be applied to the signal representing the measured value. The **Reference Signal** would therefore be the setpoint. If no reference signal is given then alarm deviation is not being used. The **Wait Time** box is for the time that you are allowing the measured value to catch the setpoint after the setpoint changes.

The **Wait Time** box under **Hold Off** is for the alarm hold off time as described at the start of this item.

Recording & Safety

Alarm recording allows you to record all alarm conditions that occur within the system into Access™ databases. There are two possible locations for these alarm recording, either in a global alarm database that holds all alarms that occur within the system, or in a batch database that can hold alarms that occur during a given batch. The dialog for configuring this is shown below.



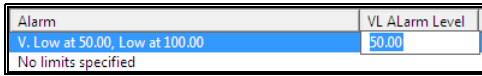
The **Alarm Database** field takes the name of the global alarm database. Although you can enter anything here, it should actually only ever say "Alarms.mdb" or be blank (blank indicates that there is no global alarm recording).

The Batch database field is different, and can be the name (or path) of any batch database. If batch alarm recording is required, then you must also enter the name of the signal within Prodigy that is being used to contain the name of the currently running batch. If batch recording is configured to use separate databases then you must check the **Separate** box on the Batch database field. This is explained elsewhere in this manual.

When an alarm occurs (within the signal under consideration) Prodigy will look at the Batch Name signal and write an alarm record into the specified batch database with the batch name forming the key part of the record. It will then also write a record into the global alarm database, if configured to do so.

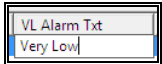
Safety Monitoring Groups should be ignored for the moment.

VL Alarm Level, L Alarm Level, H Alarm Level & VH Alarm Level



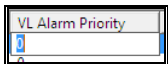
Type a new alarm level into the relevant box. If the level wasn't already in use, entering a value here will make it become used.

VL Alarm Txt, L Alarm Txt, Normal Alarm Txt, H Alarm Txt & VH Alarm Txt



Type new alarm text into the relevant box.

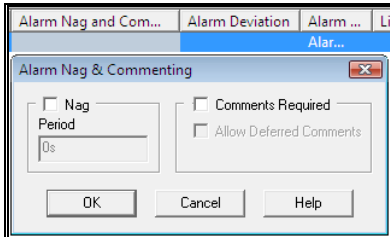
VL Alarm Priority, L Alarm Priority, H Alarm Priority & VH Alarm Priority



Type a new priority level into the relevant box.

Alarm Nag and Comments

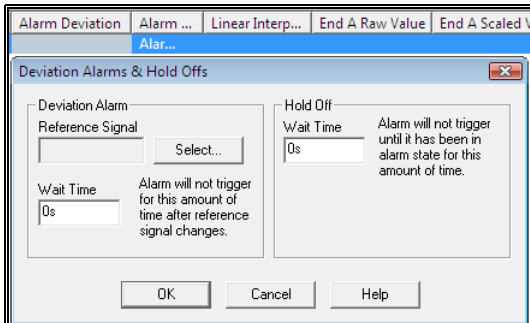
Click in the box to display the dialog for Alarm nag & Comments.



See earlier for more details about this dialog.

Alarm Deviation

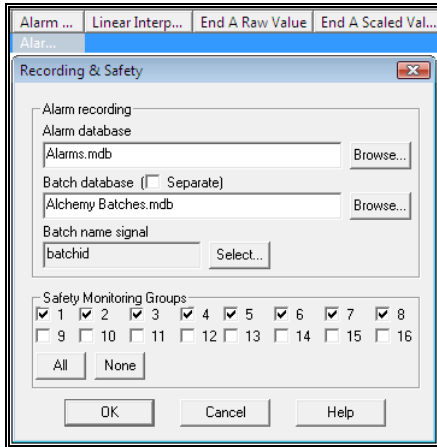
Click in the box to display the dialog for Alarm Deviation.



See earlier for more details about this dialog.

Alarm Recording

Click in the box to display the dialog for Alarm Recording.

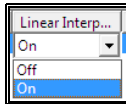


See earlier for more details about this dialog.

Linear Interpolation

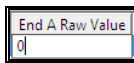
Linear interpolation determines how a value read from or written to hardware is scaled from native units to engineering units. For instance, a temperature measuring device may return a value in the range 0 to 2.5V, but this actually represents 10 to 200°C. The linear interpolation would therefore map 0V to 0°C and 2.5V to 200°C. In practice it's probably not as simple as this (0°C may map to 0.7V), but the principle remains the same.

Select whether linear interpolation is to be applied to this signal by selecting **On** or **Off** from the dropdown list. Note that linear interpolation can only be applied to **Input**, **Output** and **I/O** signals of type **Real**.



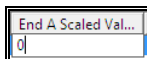
End A Raw Value

Enter the value of the raw low-end value.



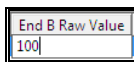
End A Scaled Value

Enter the value of the scaled low-end value.



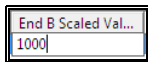
End B Raw Value

Enter the value of the raw high-end value.



End B Scaled Value

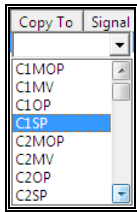
Enter the value of the scaled high-end value.



Copy To

Copy To allows you to copy the value of one signal into another. Perhaps one signal takes a global setpoint, which must be then sent down to three different controllers. One way to do that would be to have the global setpoint signal copy its value to a setpoint signal for controller A, have that signal copy its value to the setpoint signal for controller B and finally that signal copy its value to the signal for controller C.

Signals can only be copied to signals of the same type (i.e. Real to Real, Digital to Digital and so on) and cannot be copied to Input or I/O types.



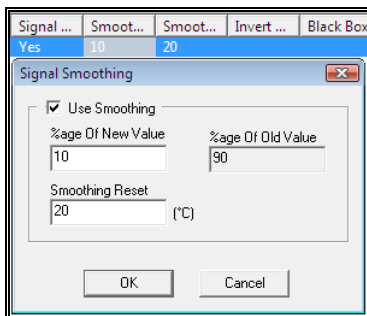
Whilst Prodigy tries to prevent "Copy To loops" (A copies to B copies to C copies to A etc.) it is not infallible and the user should take care in configuring signals with Copy To.

Signal Smoothing

Signal smoothing is used to mask the effects of noise on a signal. For instance, noise on comms line may be causing a signal to fluctuate by $\pm n\%$, leading to the controller constantly switching as it tries to attain the new (perceived) setpoint. By applying smoothing we can eliminate this constant chopping about. The smoothing applied is **exponential smoothing** whereby the signal's actual new value is set to a portion of the requested new value plus the complementary portion of the previous value. Expressed mathematically it is $V_{\text{actual}} = V_{\text{new}} * P + V_{\text{previous}} * (1 - P)$ where P is in the range 0 to 1.

If noise causes a signal at value 20 to fluctuate by ± 2 (leading to values between 18 and 22) then applying a smoothing factor of 0.1 will cause the signal to settle into fluctuations between 19.9 and 20.1, greatly reducing the amount of swing.

In our dialog the value of P is actually entered as a percentage, so 0.1 is entered as 10%.



In our example, if we set a new setpoint then the signal should move instantaneously to the new value. Smoothing would cause it to creep up slowly, which is undesirable, so the **Smoothing Reset** value is used to say, "if the value changes by this amount, jump straight to the new value". Note that this value is given in engineering units.

Smoothing can only be applied to signals of type Real and are which are either Input or I/O.

Smoothing % & Smoothing Reset

If Signal Smoothing has been enabled then you can enter the smoothing values directly into these fields by clicking in the relevant box.

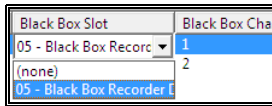
Invert Digital

Digital inversion can be applied to signals of type Digital that are either Input, Output or I/O. It does exactly as it says and inverts the value of the signal before writing it or after reading it. It's used when the value in the controller (for example) is the opposite way to what sounds sensible (e.g. 0 for "On", 1 for "Off") and helps eliminate confusion in other parts of the system.

Black Box Slot

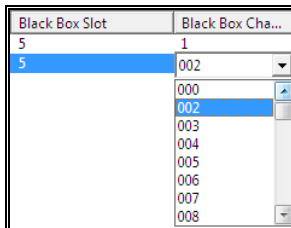
See the section in the manual about Black Box Recording for more information on this facility.

The Black Box Slot is the driver slot for the black box recorder that you want to attach this signal to. You can record up to 200 different signals in each black box recorder. Select the required driver slot from the dropdown list (below).



Black Box Channel

Each black box recorder (specified by **Black Box Slot**) can record up to 200 different signals and each signal is recorded in a channel within that recorder. This means that each channel in each slot can only have one signal allocated to it, so the dropdown list of available channels will not show channels that are already in use (below, channel 1 is already in use)



Changing Many Signals At Once

You can make changes to the same field within many rows by doing a range selection. To do this, click on the cell that you want to change on the first row that you want to change. Then shift+click in the same column but in the last row that you want to change. This will highlight all of the rows, with the selected column being a different shade. Then shift+click again in the cell you just clicked in – if it was a list-based cell the list will be displayed at that cell, otherwise a dialog will appear that allows you to enter a new value for the range of cells that you selected (below).

Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Range	Units	Event Text	Groups
38	DT1SPDIF	Machine 1 Speed Diff	Real Derived	Expression Evaluation	[DT1SPD]...	0x000...	5s	0 to 100	units/hr		Down...
50	DT2SPDIF	Machine 2 Speed Diff	Real Derived	Expression Evaluation	[DT2SPD]...	0x000...	5s	0 to 100	units/hr		Down...
62	DT3SPDIF	Machine 3 Speed Diff	Real Derived	Expression Evaluation	[DT3SPD]...	0x000...	5s	0 to 100	units/hr		Down...
79	LVLABHI	Acid Bath High Probe	Digital Derived	Expression Evaluation	[LVLABA]...	0x000...	5s	0 to 1		Off	Statu...
80	LVLABLO	Acid Bath Low Probe	Digital Derived	Expression Evaluation	[LVLABA]...	0x000...	5s	0 to 1		Off	Statu...
125	SINE01	Regular Recording	Real Derived	Signal Simulator		0x000...	5s	0 to 100	%		Internal
126	SINE02	Stepped Recording	Real Derived	Signal Simulator		0x000...	5s	0 to 100	%		Internal
127	SINE03	Intelligent Value Based	Real Derived	Signal Simulator		0x000...	5s	0 to 100	%		Internal
128	SINE04	Intelligent Slope Based	Real Derived	Signal Simulator		0x000...	5s	0 to 100	%		Internal
3	BATCHID	Batch Name	String Derived	<None>							Status
4	C1AM	Zone 1 Auto/Manual Mode	Digital Derived	<None>						Manual	Fusio...
5	C1MOP	Zone 1 Man Output Power	Real Derived	<None>						Manual	Fusio...
9	C2AM	Zone 2 Auto/Manual Mode	Digital Derived	<None>						Manual	Fusio...
10	C2MOP	Zone 2 Man Output Power	Real Derived	<None>						Manual	Fusio...
14	C3AM	Zone 3 Auto/Manual Mode	Digital Derived	<None>						Manual	Fusio...



Make changes as you would for single-cell editing. When the change is applied all of the selected cells will display the same value (below).

Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Range	Units	Event Text	Groups
38	DT1SPDIF	Machine 1 Speed Diff	Real Derived	Expression Evaluation	[DT1SPD]...	0x000...	10s	0 to 100	units/hr		Down...
50	DT2SPDIF	Machine 2 Speed Diff	Real Derived	Expression Evaluation	[DT2SPD]...	0x000...	10s	0 to 100	units/hr		Down...
62	DT3SPDIF	Machine 3 Speed Diff	Real Derived	Expression Evaluation	[DT3SPD]...	0x000...	10s	0 to 100	units/hr		Down...
79	LVLABHI	Acid Bath High Probe	Digital Derived	Expression Evaluation	[LVLABA]...	0x000...	10s	0 to 1		Off	Statu...
80	LVLABLO	Acid Bath Low Probe	Digital Derived	Expression Evaluation	[LVLABA]...	0x000...	10s	0 to 1		Off	Statu...
125	SINE01	Regular Recording	Real Derived	Signal Simulator		0x000...	10s	0 to 100	%		Internal
126	SINE02	Stepped Recording	Real Derived	Signal Simulator		0x000...	10s	0 to 100	%		Internal
127	SINE03	Intelligent Value Based	Real Derived	Signal Simulator		0x000...	10s	0 to 100	%		Internal
128	SINE04	Intelligent Slope Based	Real Derived	Signal Simulator		0x000...	5s	0 to 100	%		Internal
3	BATCHID	Batch Name	String Derived	<None>			1s				Status
4	C1AM	Zone 1 Auto/Manual Mode	Digital Derived	<None>			1s	0 to 1		Manual	Fusio...
5	C1MOP	Zone 1 Man Output Power	Real Derived	<None>			1s	0 to 100	%		Fusio...
9	C2AM	Zone 2 Auto/Manual Mode	Digital Derived	<None>			1s	0 to 1		Manual	Fusio...
10	C2MOP	Zone 2 Man Output Power	Real Derived	<None>			1s	0 to 100	%		Fusio...
14	C3AM	Zone 3 Auto/Manual Mode	Digital Derived	<None>			1s	0 to 1		Manual	Fusio...
15	C3MOP	Zone 3 Man Output Power	Real Derived	<None>			1s	0 to 100	%		Fusio...


Copying The Attributes Of One Signal To Another

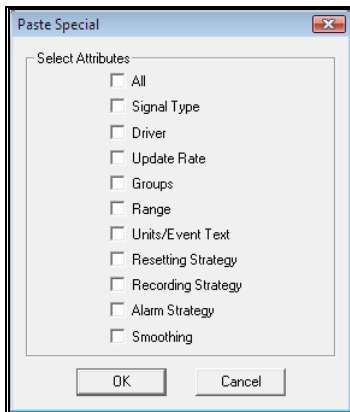
If you have one signal with a particular set of attributes and you want to copy these attributes to another signal (or signals) you can do this easily. Click on the signal whose attributes you want to copy and click the **Copy** button



. Then click on the signal that you want to copy attributes to and click the **Paste** button




If you don't want to copy all attributes, press the **Paste Special** button  instead of the Paste button. This will display a dialog that allows you to choose the attributes that you want to paste.



Check the boxes of the attributes you want to paste and then click OK.

Deleting Signals

To delete a signal or signals click on the signal(s) to be deleted and then click the **Delete** button . If you have selected more than one signal you will be prompted to confirm that you do want to delete all of the signals.

Navigating The Signal Database – Filtering, Searching, Etc.

There is a very powerful and comprehensive set of filtering tools within Signal Database Configuration. These allow you to work easily within the largest of signal databases, focussing on the signals you are interested in and eliminating the clutter.



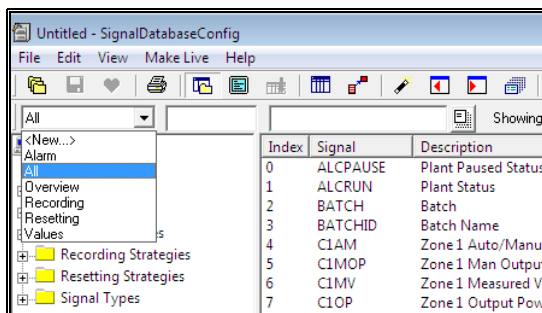
At the top of the display is a toolbar that contains search and selection options, as well as a status line. The status line shows how many signals are in the database, how many can be seen in the filtered view, how much room is left in the signal database, and how many I/O tags have been used. You should keep an eye on the section that shows how much room is left in the signal database – it may become necessary to increase the database size as you progress. If so, it is usually better to do this at a convenient moment than to have the change forced upon you when you find you cannot perform the operation that you are attempting (e.g. creating a block of signals). See later in the manual for how to do this.

The boxes on the toolbar are as follows (from left to right):

- **Column Presets.**
This drop down list contains predefined and custom column lists.
- **Signal Name Search/Filter.**
When you start to type into this field the system will search for a signal that begins with the characters that you type and highlight the first row that matches.
- **Signal Description Search/Filter.**
When you type into this field the system will search for a signal description that contains the string that you type. It will highlight the first row that matches.
- **Search/Filter Toggle Button.**
When you click this button the system filters by the strings entered in the two boxes rather than searching. Unclicking the button reverts to search mode.

Column Presets

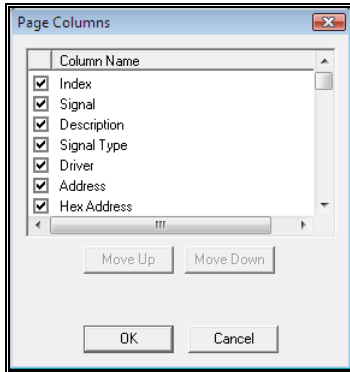
The set of columns that is displayed on the grid is not fixed. The user can choose from a list of predefined sets and they can create their own custom sets. To select a different column set from the list of predefined and custom made options, simply choose the required set from the drop-down list in the top-left corner of the display.



The columns displayed on the main portion of the screen will change to reflect the column set that was selected.



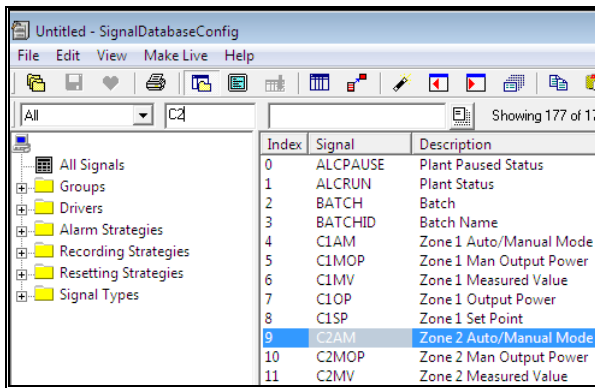
To customise the view click on the “Columns” button. This displays a dialog that allows you to select which columns to display (check or uncheck the boxes) and in which order to display them (use the “Move Up” and “Move Down” buttons).



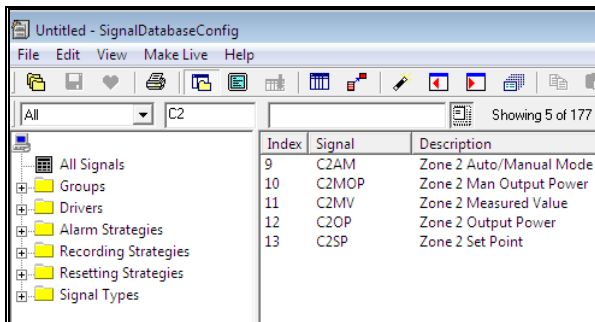
If you want to save this column set as a custom configuration, select “<New...>” from the drop-down list in the top-left corner. This will display a dialog asking for the name of the new column set. This column set will be saved and will be available each time that you use Signal Database Configuration in the future. If you change the columns and/or their order whilst using a custom set, those changes will be automatically saved and made permanent. If you make changes to one of the predefined sets (Alarm, All, Overview, Recording, Resetting and Values) these changes will be lost when you exit the program – changes to the predefined set *cannot* be made permanent.

Signal Name Search/Filter

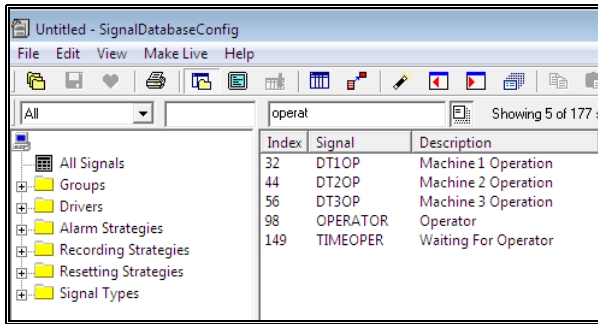
Type the first few characters of the signal (or group of signals) that you want to search for into the box. The row highlighter will move to the first row that contains a match. For best results you should order the columns on Signal Name (by clicking on the “Signal” column header).



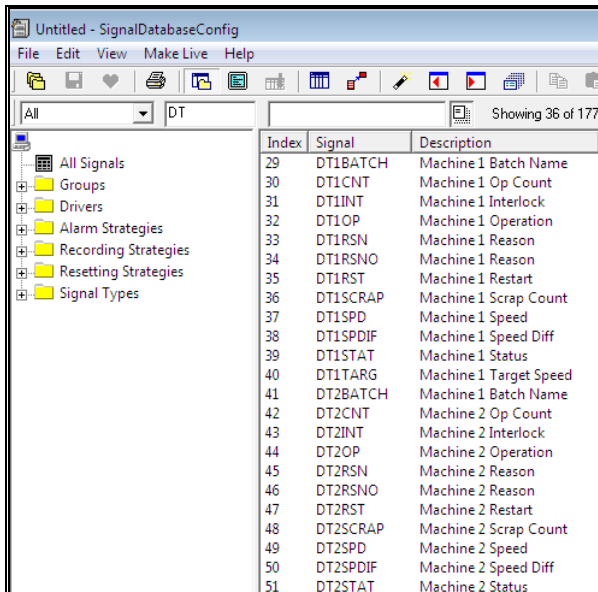
If you click the Search/Filter toggle button, this will switch to a filtered view containing only those signals that begin with the text you have typed.



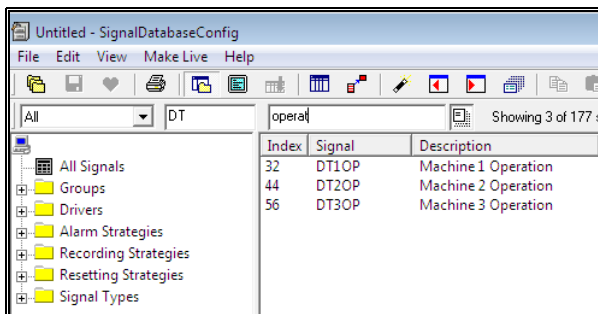
The same happens with the description search/filter, except that the text you type can occur anywhere within the description, not just its start.



The two can work in conjunction with each other. Filtering only on Signal Name...

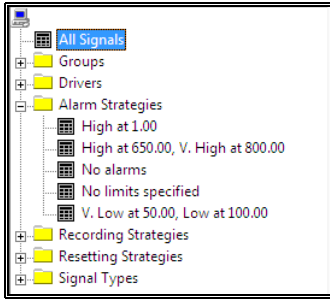


Filtering by Signal Name *and* Description...



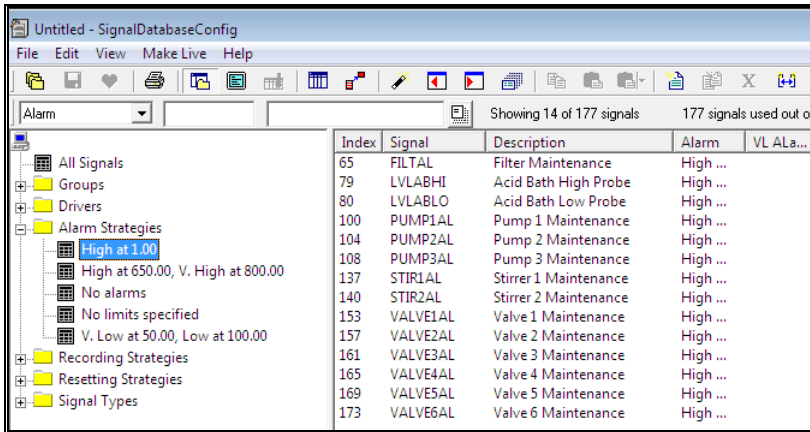
Filtering By Signal Attributes

Down the left hand side of the display are a group of signal attribute filters. These are grouped together by type, and each grouping contains all of the unique instances of that particular attribute within the signal database. For instance, a number of signals may have been given the same Alarm Strategy and a number of other signals may have been given another, different strategy. The Alarm Strategies filtering group would therefore contain two entries. Clicking on either of them would result in only those signals that use that strategy being displayed. Clicking on the other would now switch the filter to show those signals that use *that* strategy. Below is the Alarm Strategies grouping opened up for our example database.

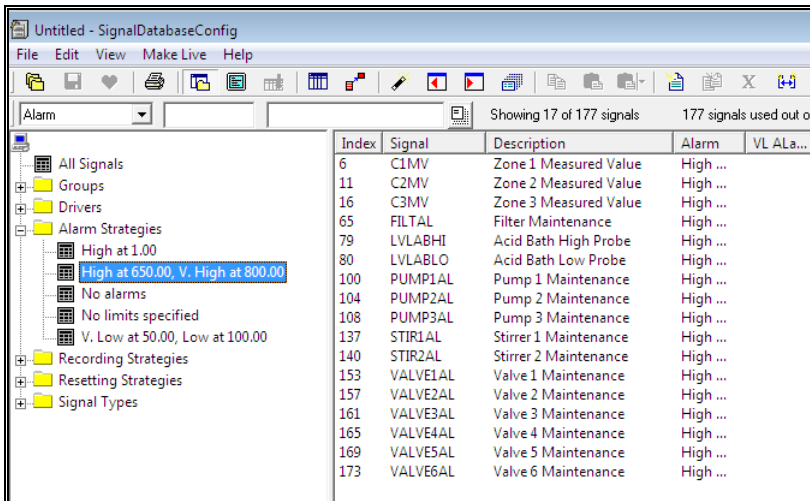


There are four different Alarm Strategies used within our signal database and in addition some signals have no alarms.

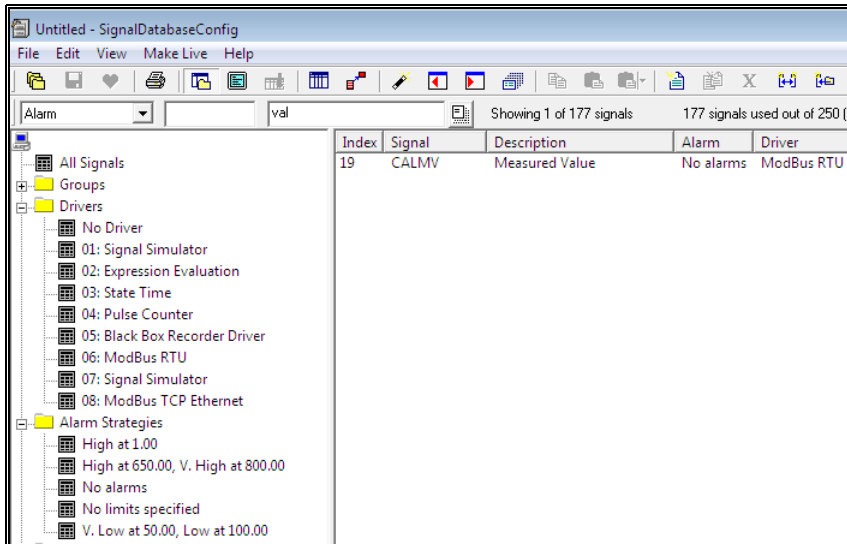
If we click on one of the entries here, the system will filter to show only those signals that have the Alarm Strategy we click on (below).



If we shift+click on another Alarm Strategy this applies an **Or** type of operation. The system will show all signals that use **either** of those strategies.

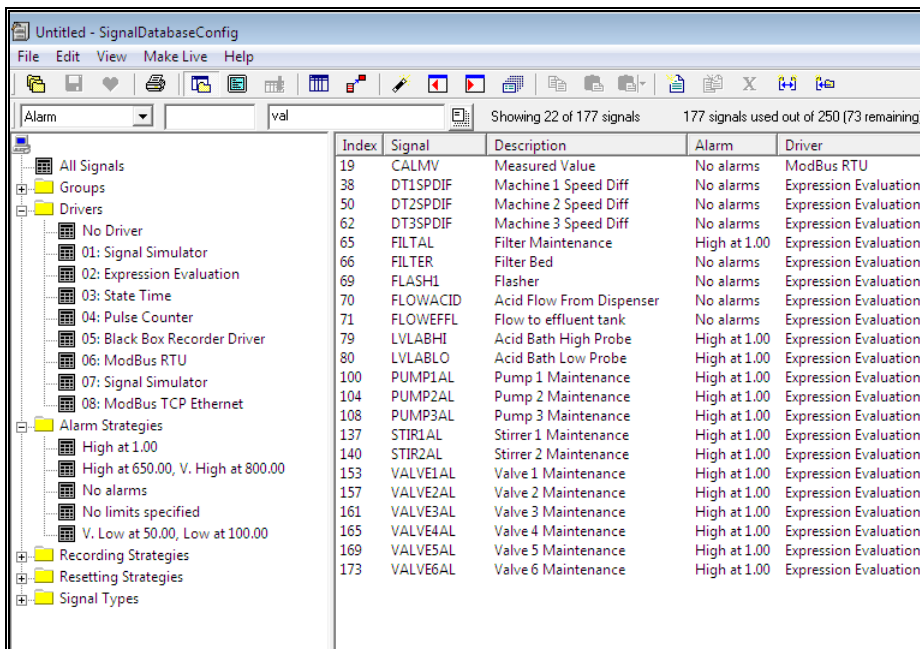


We can also click on items from inside other groups. This applies an **And** type of operation and the system will show all signals that have (for example) the required alarm strategy **and** the required driver.

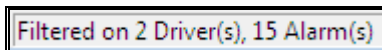


The example above is filtered to show all signals with no alarm strategy that use the Modbus RTU driver (and there's only one).

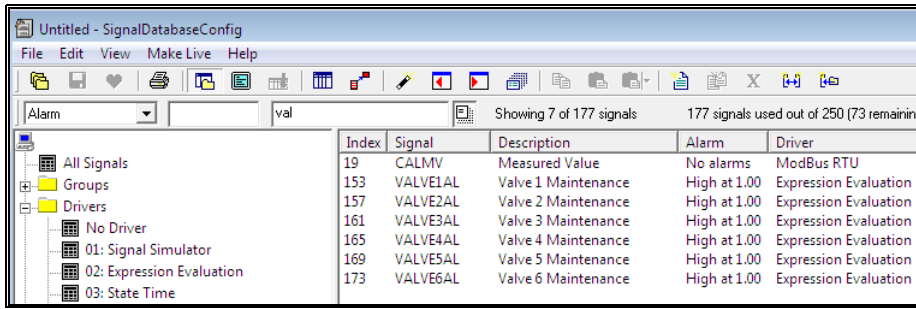
Remember, shift+clicking *within* a group does an **Or** for the things in that group. Clicking in *other* groups does an **And** between the different groups. So for example we can select all signals that have either no alarm strategy or are "High at 1.0" and which use either the Modbus RTU driver or the Expression Evaluation driver (below).



There is a Status Bar at the bottom of the display that shows how many of each group you are filtering on (below).



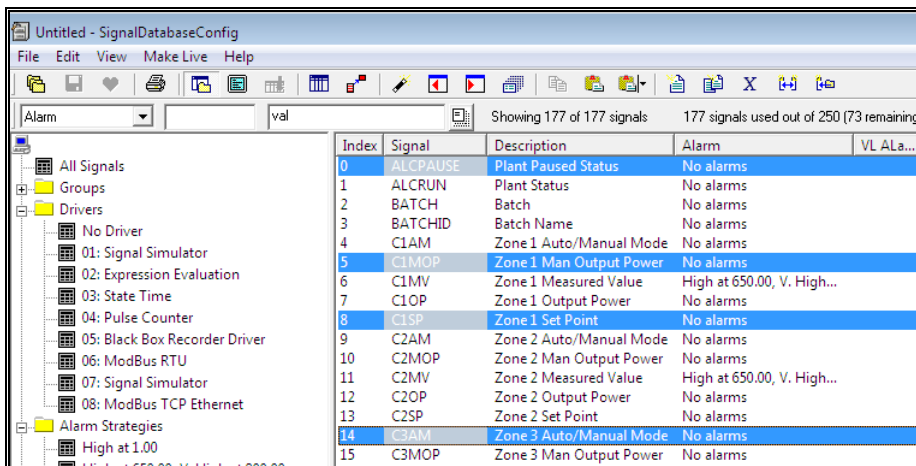
Again, these filtering options work in conjunction with the Signal Name & Description filtering described earlier. Below is the database filtered as above but in addition showing only signals whose description contains "val".



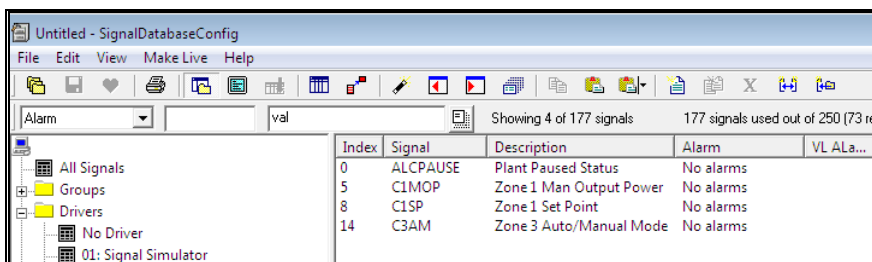
To revert to showing all signals click the All Signals option at the top of the filtering groups and (if necessary) clear any text from the Signal Name & Description fields.

Filtering By Selection

The final way that you can filter is to manually select the signals you want to view and choose to see only those signals. Click on one signal and then shift+click or ctrl+click on subsequent signals to make your selection (below).




Click the **Show Selection Only** button  to filter on only these signals (below).

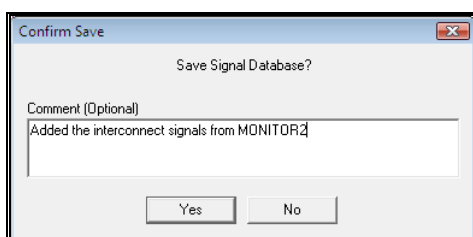


Saving The Signal Database

There are two ways you can save the signal database; you can simply save it or you can “save and go live”.

Simply Saving


Click the **Save** button on the toolbar . A dialog will be displayed that allows you to label this version of the signal database that you are about to save.



This is most useful should you wish to go back to a previous version of the signal database (because you made a mistake, or the system doesn't work as expected, or so on). Enter a comment and click the Yes button to save the signal database.

Save And “Go Live”

Simply saving the signals database does not mean that the things you have changed will take immediate effect. All that happens is that the signal database is written away to disk. The next time that Prodigy is restarted it will read this signal database and at that point the changes that you have made will begin to be used. This is not always convenient, especially in control applications with long run times. It may be days, weeks or even months before it is convenient to restart Prodigy. Instead you can use a facility called **Go Live** which applies all of the changes that you have made to the database that is currently being run (there are some caveats to this that will be discussed later).

To Go Live click the  button. If you have not already saved the signal database you will be prompted to do so and to give a comment (as described previously). The system will go through the stages necessary to apply all of your changes to the currently running database (a status dialog will appear showing these stages as they occur).

Things that can happen include:

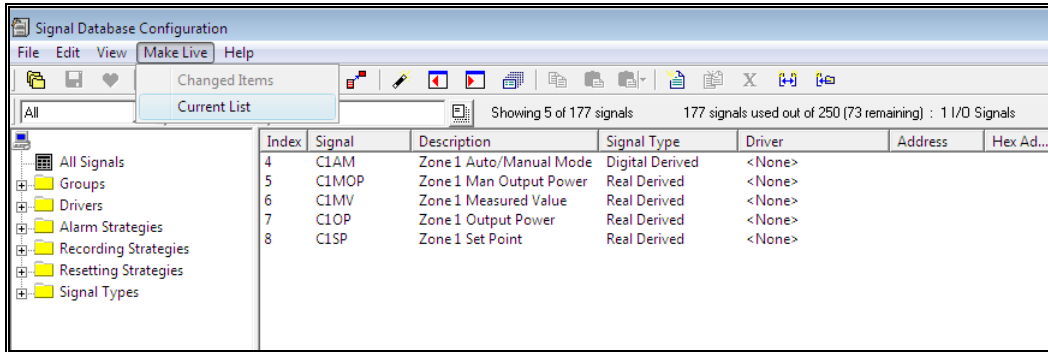
- Changes to simple signal attributes (descriptions, units, ranges etc.) are applied.
- New signals are added.
- Deleted signals are marked as deleted (they are not actually deleted at this point, that will occur when Prodigy is next restarted).
- Changes to driver attributes are applied. These may require that the driver is restarted, or if the driver was not already running that it is started now. Be careful when making changes that it is OK for a driver to be restarted. The pause will only be momentary but for a while part of a system may not be being controlled while the driver is restarted.
- Changes to managed attributes (e.g. alarm, recording and resetting strategies) are applied. These may necessitate that another part of the Prodigy system reinitialises or restarts itself. Signal Database Configuration ensures that this happens.

The caveats as promised earlier are:

- If the newly saved database has more signals in it than can be accommodated by the current live database, the newly created signals cannot be made live. This is why you should always ensure there is a degree of spare capacity within your live database, unless you are sure that you will not want to add more signals “on the fly”.
- Whilst Signal Database Configuration can add and start newly added drivers, it will not stop and/or remove drivers that are no longer used.


Selective “Go Live”

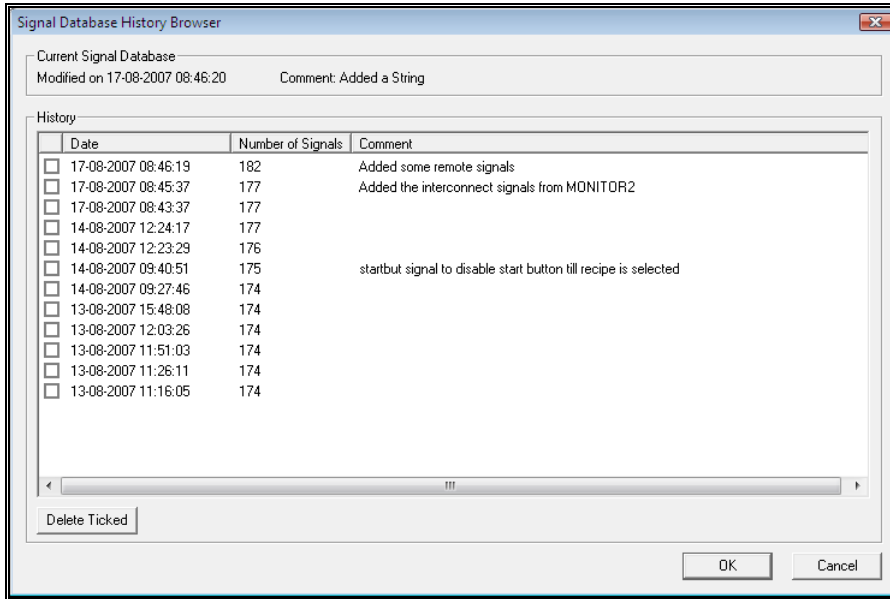
It is possible in signal database to make a selection of signals live; this is a useful feature if you have created lots of new signals but only wish to test or use a few of them at the present time without activating all of the other signals. To make a selection of signals live you must filter the required signals, how this is done was described in navigating the signal database. Once the selection of signals has been made click on the make live drop down menu and select current list as shown below this will then make the selected signals live.



Disaster Recovery

Loading From The History File

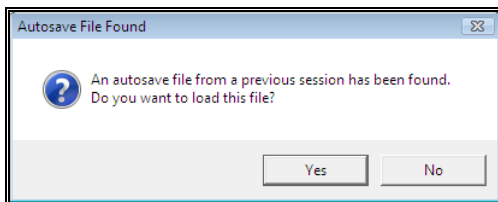
Every time you save the signal database, the previous version is copied to a history folder. This means that at any time you can retrieve a previous version of the signal database and restore it as the current database. To do this click on the **History** button . This will display a dialog that shows all of the previously saved signal databases (or at least those that remain – you can delete unwanted ones, see later).



You can see now why the comments you enter when saving the database are useful. They label that version of the signal database so that you can identify it in the history browser. For added convenience the date that the history entry was made is also shown. To load in a signal database from the history click on the one you require and then click **OK**. The program will load in the selected signal database to become the current working database. Note that the existing database *in memory* will be lost, so save any changes you have made first, if necessary. The loaded history database will not become the current database on disk until you hit the **Save** button. At this point, as always, you will be prompted to enter a comment and the current database on disk will be moved to the history folder. Perhaps you should enter something like “Restored database from history because...”

Loading From AutoSave

Every 30 seconds the program performs an AutoSave of the current working database as it stands *in memory*. This save takes place as a background operation and will go unnoticed by you (for efficiency’s sake it does not perform this save if you have not made any changes to the database). If the Signal Database Configuration program should crash (or be killed of for any other unforeseen reason) the AutoSave file will be left behind. When you start the program again it will detect this and display the following dialog.



If you do want to load the AutoSave file, click the **Yes** button. When the program loads the AutoSave file the signal database will be in a unsaved state, as if you had made changes.

If, in the process of adding signals prior to the crash that left behind the AutoSave file, you had added new drivers, this fact will be lost. You will have to start the drivers manually (and maybe even add them to the driver list) because the program will not now do it for you.

Exiting the program will cause the AutoSave file to be deleted, indicating normal program termination.

Signal Database Import/Export

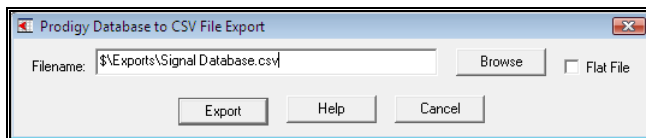
The Signal Database Import/Export functions exist to aid in the production of large signal databases where there are many similar signals to enter. The existing signal database is exported to a comma separated variable format file which can be read in and edited in Excel. The resulting file is then imported back into Prodigy, and replaces the existing signal database.

How to use the Import/Export facility

Create the Initial Database

It is easiest to copy/paste the details of existing signals in Excel, rather than enter all the data by hand, so create an example of each type of signal you will wish to copy in **Signal Database Configuration** first. For instance if you had a signal database that was to consist of 100 modbus digital inputs, 100 modbus analogue inputs, 10 modbus analogue outputs and 20 derived digital signals, you should enter 1 of each of these in the normal way through **Signal Database Configuration**. Set up all the details, including alarms, groups, etc.

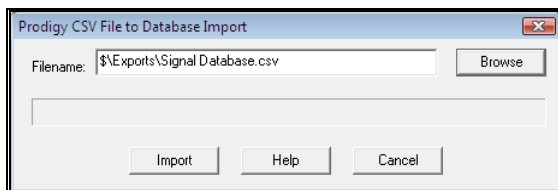
Export the Signal Database.



Use the Signal Database Export facility, which displays the dialog shown above. Use the **Browse** button to select the directory and file into which the database is to be exported. The database can be exported in one of two ways, either as a flat file or as tables. If **Flat File** is selected each signal has all of its information on one line, and all information is unique to each signal. If Flat File is not selected then signals retain pointers into groups of auxiliary data, such as recording strategies, allowing several signals to access the same table entry (as described in Signal database Configuration elsewhere in this manual).

Next edit the database in Excel™ (or some other spreadsheet). Copy and paste each signal type to make more of that type of signal, then edit the columns that differ between the signals (e.g. name, address). Some addresses can be split up into a more user-friendly format than the combined address – the details of these are explained later.

Import the Database Back Into Prodigy.



Warning: Importing a database *replaces* the existing database, therefore you *must* back up your system prior to importing a database.

Friendly Addressing

Modbus

Addr03 = Device Number

Addr04 = Parameter Type

Addr05 = Parameter Address

Addr06 = Bit Number

MelsecA

Addr03 = PC Number

Addr04 = Station Number


Addr05 = Type

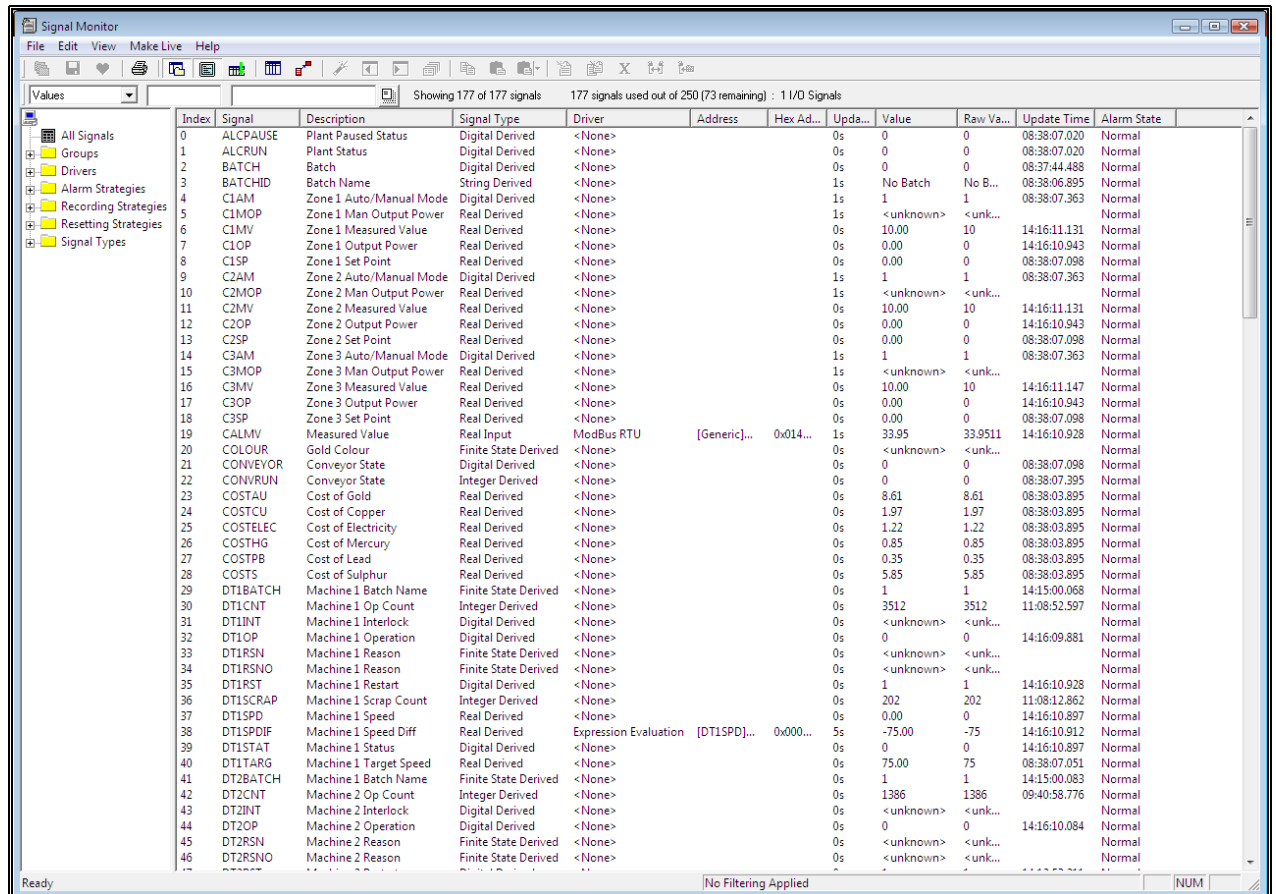
Addr06 = Address

Signal Monitor

Signal Monitor

Signal Monitor is a quick and easy way to check and set your signal values or edit alarm properties. It is part of the

Signal Database Configuration program and can be activated by pressing the signal monitor button  below is a view of Signal Database Configuration when in monitor mode, notice the signal monitor button is indented. (Signal Monitor can also be run as a separate program that can accessed via the menu bar)



Index	Signal	Description	Signal Type	Driver	Address	Hex Ad...	Upda...	Value	Raw Va...	Update Time	Alarm State
0	ALCPAUSE	Plant Paused Status	Digital Derived	<None>			0s	0	0	08:38:07.020	Normal
1	ALCRUN	Plant Status	Digital Derived	<None>			0s	0	0	08:38:07.020	Normal
2	BATCH	Batch	Digital Derived	<None>			0s	0	0	08:37:44.488	Normal
3	BATCHID	Batch Name	String Derived	<None>			1s	No Batch	No B...	08:38:06.895	Normal
4	CIAM	Zone 1 Auto/Manual Mode	Digital Derived	<None>			1s	1	1	08:38:07.363	Normal
5	CIMOP	Zone 1 Man Output Power	Real Derived	<None>			1s	<unknown>	<unk...		Normal
6	CIMV	Zone 1 Measured Value	Real Derived	<None>			0s	10.00	10	14:16:11.131	Normal
7	C1OP	Zone 1 Output Power	Real Derived	<None>			0s	0.00	0	14:16:10.943	Normal
8	C1SP	Zone 1 Set Point	Real Derived	<None>			0s	0.00	0	08:38:07.098	Normal
9	C2AM	Zone 2 Auto/Manual Mode	Digital Derived	<None>			1s	1	1	08:38:07.363	Normal
10	C2MOP	Zone 2 Man Output Power	Real Derived	<None>			1s	<unknown>	<unk...		Normal
11	C2MV	Zone 2 Measured Value	Real Derived	<None>			0s	10.00	10	14:16:11.131	Normal
12	C2OP	Zone 2 Output Power	Real Derived	<None>			0s	0.00	0	14:16:10.943	Normal
13	C2SP	Zone 2 Set Point	Real Derived	<None>			0s	0.00	0	08:38:07.098	Normal
14	C3AM	Zone 3 Auto/Manual Mode	Digital Derived	<None>			1s	1	1	08:38:07.363	Normal
15	C3MOP	Zone 3 Man Output Power	Real Derived	<None>			1s	<unknown>	<unk...		Normal
16	C3MV	Zone 3 Measured Value	Real Derived	<None>			0s	10.00	10	14:16:11.147	Normal
17	C3OP	Zone 3 Output Power	Real Derived	<None>			0s	0.00	0	14:16:10.943	Normal
18	C3SP	Zone 3 Set Point	Real Derived	<None>			0s	0.00	0	08:38:07.098	Normal
19	CALMV	Measured Value	Real Input	ModBus RTU	[Generic]...	0x014...	1s	33.95	33.9511	14:16:10.928	Normal
20	COLOUR	Gold Colour	Finite State Derived	<None>			0s	<unknown>	<unk...		Normal
21	CONVEYOR	Conveyor State	Digital Derived	<None>			0s	0	0	08:38:07.098	Normal
22	CONVRUN	Conveyor State	Integer Derived	<None>			0s	0	0	08:38:07.395	Normal
23	COSTAU	Cost of Gold	Real Derived	<None>			0s	8.61	8.61	08:38:03.895	Normal
24	COSTCU	Cost of Copper	Real Derived	<None>			0s	1.97	1.97	08:38:03.895	Normal
25	COSTELEC	Cost of Electricity	Real Derived	<None>			0s	1.22	1.22	08:38:03.895	Normal
26	COSTHG	Cost of Mercury	Real Derived	<None>			0s	0.85	0.85	08:38:03.895	Normal
27	COSTPB	Cost of Lead	Real Derived	<None>			0s	0.35	0.35	08:38:03.895	Normal
28	COSTS	Cost of Sulphur	Real Derived	<None>			0s	5.85	5.85	08:38:03.895	Normal
29	DTIBATCH	Machine 1 Batch Name	Finite State Derived	<None>			0s	1	1	14:15:00.068	Normal
30	DTICNT	Machine 1 Op Count	Integer Derived	<None>			0s	3512	3512	11:08:52.597	Normal
31	DTIINT	Machine 1 Interlock	Digital Derived	<None>			0s	<unknown>	<unk...		Normal
32	DTIOP	Machine 1 Operation	Digital Derived	<None>			0s	0	0	14:16:09.881	Normal
33	DTIRSN	Machine 1 Reason	Finite State Derived	<None>			0s	<unknown>	<unk...		Normal
34	DTIRSNO	Machine 1 Reason	Finite State Derived	<None>			0s	<unknown>	<unk...		Normal
35	DTIRST	Machine 1 Restart	Digital Derived	<None>			0s	1	1	14:16:10.928	Normal
36	DTISCRAP	Machine 1 Scrap Count	Integer Derived	<None>			0s	202	202	11:08:12.852	Normal
37	DT1SPD	Machine 1 Speed	Real Derived	<None>			0s	0.00	0	14:16:10.897	Normal
38	DT1SPDIF	Machine 1 Speed Diff	Real Derived	Expression Evaluation	[DT1SPD]...	0x000...	5s	-75.00	-75	14:16:10.912	Normal
39	DT1STAT	Machine 1 Status	Digital Derived	<None>			0s	0	0	14:16:10.897	Normal
40	DT1TARG	Machine 1 Target Speed	Real Derived	<None>			0s	75.00	75	08:38:07.051	Normal
41	DT2BATCH	Machine 1 Batch Name	Finite State Derived	<None>			0s	1	1	14:15:00.083	Normal
42	DT2CNT	Machine 2 Op Count	Integer Derived	<None>			0s	1386	1386	09:40:58.776	Normal
43	DT2INT	Machine 2 Interlock	Digital Derived	<None>			0s	<unknown>	<unk...		Normal
44	DT2OP	Machine 2 Operation	Digital Derived	<None>			0s	0	0	14:16:10.084	Normal
45	DT2RSN	Machine 2 Reason	Finite State Derived	<None>			0s	<unknown>	<unk...		Normal
46	DT2RSNO	Machine 2 Reason	Finite State Derived	<None>			0s	<unknown>	<unk...		Normal

The fields that are shown are customisable, as is the order in which they are shown. When the **Value** field is shown the current value of the signal is displayed here and is constantly refreshed as the value changes in the signal database.

Navigating – Filtering, Searching, Etc.

Signal monitor mode possesses all the same features for filtering and searching that the signal database configuration program has. How to use these features can be found in the relevant section of the **Signal Database Chapter**. When the program is initially run up it looks as above, showing default signal fields for all signals in the database.

Altering Signal Properties

Right clicking on a row will allow you to change some properties for the signal that that row relates to. The following context menu appears:

Signal	Description	Signal Type
ALCPAUSE	Alarm Disabled Feature	Digital Derived
ALCRUN	Signal Value	Digital Derived
BATCH	Alarm Properties	Digital Derived
BATCHID		String Derived
CIAM	Delete	Digital Derived
CIMOP		Real Derived
C1MV	Copy	Real Derived
C1OP	Paste	Real Derived
C1SP	Paste Special...	Real Derived
C2AM		Digital Derived
C2MOP	Find & Replace	Real Derived
C2MV	Zone 2 measured value	Real Derived
C2OP	Zone 2 Output Power	Real Derived

It can be seen that Alarm Properties is greyed out, this is because the signal ALCPAUSE does not have any alarm limits so they cannot be altered; below is the context menu when a signal with alarm limits has been selected. As a user within signal monitor you cannot add alarm limits to signals that don't have any already, you can only alter the alarm limits of signals that do.

Description	Signal Type	Driver
Acid Dispenser	Digital Derived	None
Copper Level	Signal Value	None
Effluent Level	Alarm Properties	None
Fusion Reactor Level		None
Mercury Level	Delete	None
Ingredients Hopper Level	Copy	None
Lead Level	Paste	None
Sulphur Level	Paste Special...	None
Separator Level	Find & Replace	None
Maint Icons Visibility		None
Network Flasher 1	Integer Derived	None
Network Flasher 2	Integer Derived	None

Select which type of signal property you want to alter – you may require a privilege for whichever option you choose.

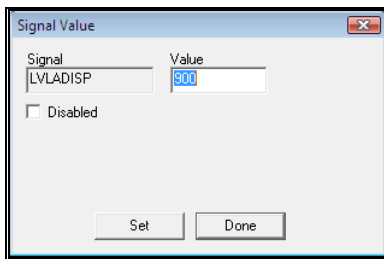
Selecting **Alarm Properties** displays the following dialog:

If the alarm strategy of the selected signal is also used by other signals a warning dialog will have been displayed first.

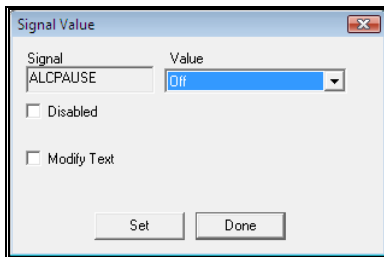
Change any of the fields on this dialog as required. Pressing the **Alarm Nag & Commenting** or **Deviation & Hold Offs** buttons will display the same dialogs as described in the relevant section of the **Signal Database Configuration** chapter.

You can alter the values on this dialog and apply them to the live database by pressing **OK**. At this point you may also be asked if you want to save the changes to the disk file, so that they will remain in effect after a restart. This again is privilege dependent, the appropriate dialog can be seen below.

Selecting **Signal Value** from the menu will display the following dialog, if you obtain the relevant privilege:



The dialog will differ slightly depending on the **type** of the signal. The example above is for an **analogue** signal, that below is for a **digital**.



You can alter or **Disable** the signal value from this dialog.

Sorting

The data can be sorted on any column simply by clicking the header of that column. Clicking the column again will reverse the sort order for that column.

Printing

A report of the signal list can be printed by pressing the **Print** button. The selected fields of the currently displayed signals are printed (much as it looks on-screen) and a list of which filters are applied is also printed in the header of the report.

Display Builder

What Display Builder Can Do

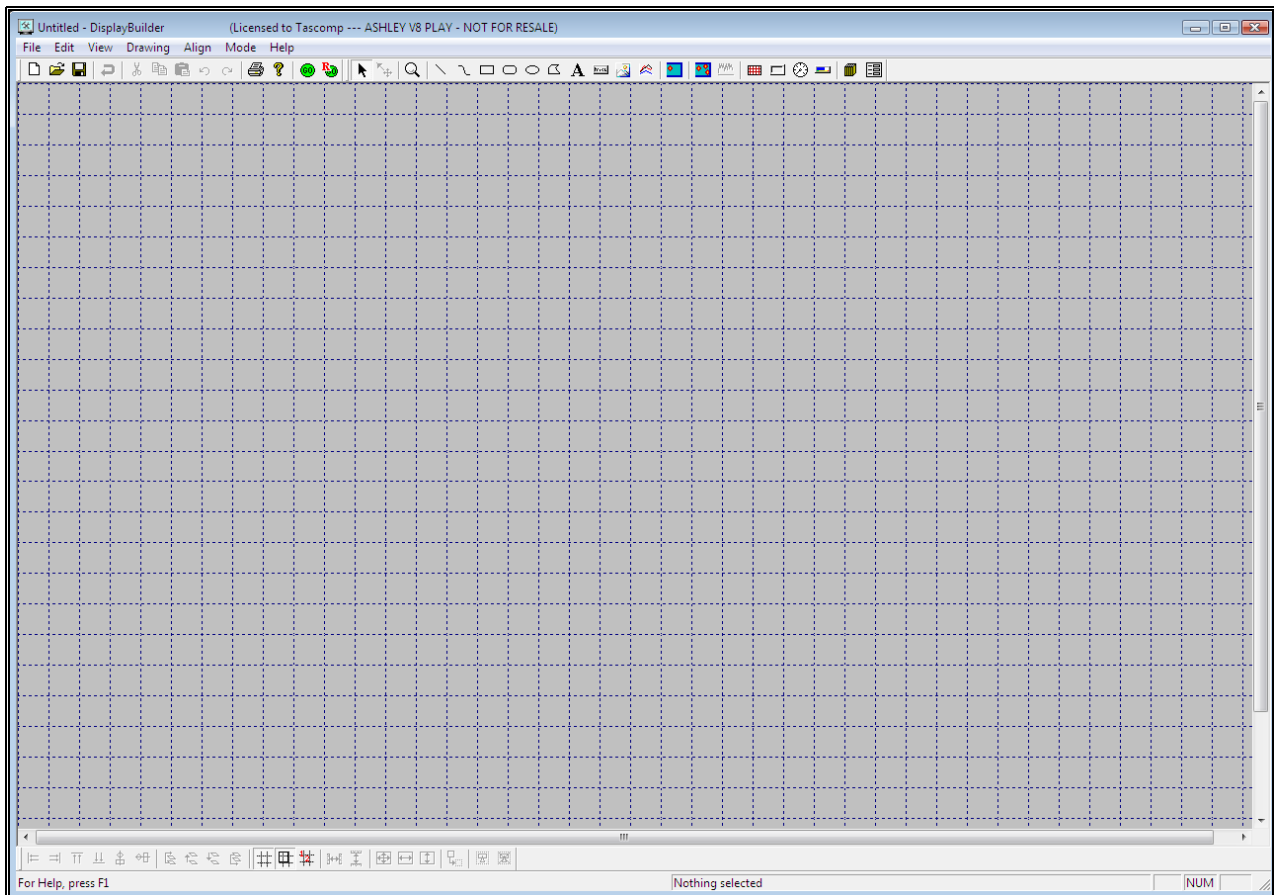
Display Builder's purpose is to show a graphical representation of your process, enabling the operators to quickly determine its current state.

Each item in a Prodigy Display Builder display is an object and any number of these objects can be used to build a display. Each object has a set of properties that can be affected by changes in signal value and these properties allow almost every aspect of an object's appearance to be altered. For instance, a change in signal value may make an object move, grow, rotate, change colour or vanish. In addition, certain object types have properties that allow signal values to be set, whether these signals are integer, analogue or digital.

Prodigy Display Builder also has an object type for the real-time and historical trending of data, and for the construction, retrieval and application of recipe and batch data.

How Do You Construct A Display Builder Display?

When you invoke Display Builder Edit you are presented with a blank sheet on which to begin constructing your Display.



Display Builder objects are then placed on the Display by selecting the required object from the **Drawing Toolbar** and dragging it out onto this blank area.

The Drawing Toolbar

The toolbar at the top of the screen is the **Drawing Toolbar** and gives access to all of the object types available.



Reading from right to left, the functions offered by the drawing toolbar are

Selection.



This allows any objects already placed on the screen to be selected, enlarged or moved.

Point Edit



This allows individual points within certain objects (e.g. lines, curves) to be moved around.

Zoom.



Allows the user to zoom in on an area of a Display. Either single click over an object to zoom in on that object or click and drag a box to zoom in on the area enclosed by that drag box. To zoom out in increments right click over the Display area. To go straight back to a 1 to 1 view click on the Zoom icon again.

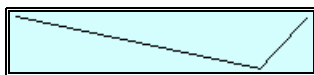
When the zoom tool is selected the cursor changes into a magnifying glass which shows “+/-” when it is possible to zoom in or out, “+” when it is only possible to zoom in and “-” when it is only possible to zoom out.

Line(s).



Allows a series of connected lines to be drawn on the display, all of the lines in a given object sharing the same properties. Note that closed polygons are not constructed by this function. Click at the point where the first line segment is to begin, then click (do not drag) at the next point, and the next and so on. Line segments continue to be added until a double click is performed to indicate that the last line segment has been added.

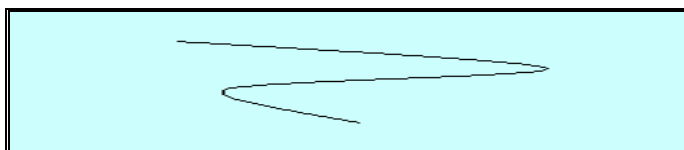
Extra nodes can be added into a line by double clicking on a segment. A new node will be added half way between the node points at either end of the line segment that you click on.



Curve(s)



Allows a series of curves to be described. The curves are drawn using point fitting rather than bezier methods and curve segments are added in the same way as line segments described above.



Rectangle.



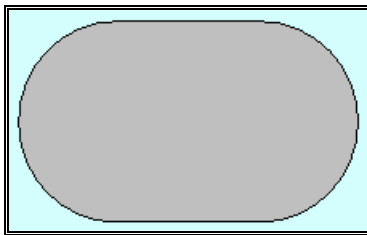
Allows a rectangle (or square) to be drawn on the display.



Rounded Rectangle.



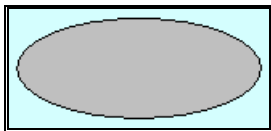
Allows a ‘rounded rectangle’ i.e. a rectangle with rounded corners and/or edges to be added to the display.



Ellipse.



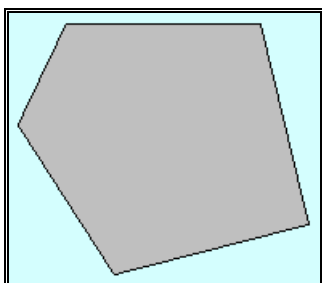
Allows an ellipse (or circle) to be added to the display.



Closed Polygon.



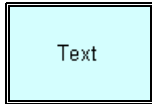
Allows a series of connected lines to be added to the display, the first point always being connected to the last to automatically form a closed polygon. The polygon is drawn in closed form as it is being constructed and will be filled in a default colour. Line segments continue to be added until a double click is performed to indicate that the last segment has been added.



Text



Allows static text or the value of a signal to be displayed.



Button.



This allows a “real” windows button to be created on the Display Builder display. This button may be of any of the windows button styles – pushbutton, check box or radio button.



Bitmap.



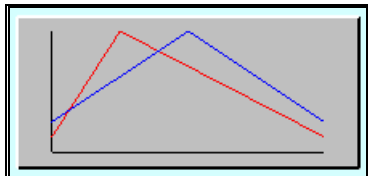
Allows a bitmap file to be imported onto the display. This file may be constructed by any package supporting the .BMP file format, and should be a 256 colour bitmap image.



Trend.



Allows an object to be added to the display that will provide real-time and/or historical trending of signals. Through the use of a historical Trend object, access is also gained to the data export facility.



Heatmap.



This allows a heatmap object to be created. Heatmap objects are for use with the Raytek heat scanner device driver and are not available if your license does not cover this object.

Advanced Heatmap



This allows an advanced heatmap object to be created. Heatmap objects are for use with the Raytek heat scanner device driver and are not available if your license does not cover this object

Heatmap Extra



This allows extra data-readoff facilities to be added to an Advanced Heatmap object.

Grid



This allows a grid object to be placed onto the display. A grid is a sequence of horizontal and/or vertical lines spaced evenly over the area of the grid object.

Frame



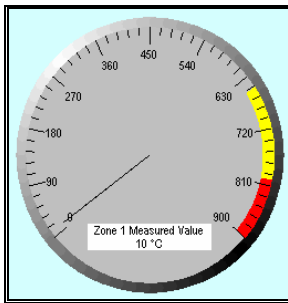
This allows a Frame object to be created. A Frame consists of a 3-D bezel and a solid background. The colours of all aspects of the frame can be customised and the background can be dynamic.



Dial



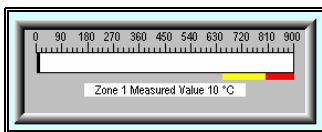
This allows a dial object to be created. Dials are quite sophisticated display objects with a lot of customisable properties. They are covered in more detail later in this section.



Bar Graph



This allows a Bar Graph object to be created. Bar Graphs are quite sophisticated display objects with a lot of customisable properties. They are covered in more detail later in this section.



Intelligent Objects.



Selecting this toolbar button provides a further toolbar containing several categories of intelligent object. An intelligent object is an object that has been constructed from several basic objects and assigned a set of properties that interact correctly to provide the relevant functionality for that type of composite object, for instance a “speedometer-type” dial.

Forms.



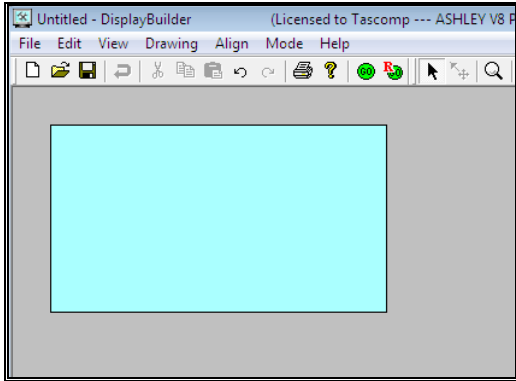
This allows forms to be designed that enable the user to create and maintain recipe information, most likely governing signal setpoint values, and also forms to allow batch parameters to be entered. Once designed, a form may be invoked by setting the relevant button property of a basic or intelligent object.

How Do You Place an Object?

Select the relevant object from the toolbar, for example a rectangle, by pressing the toolbar button.



Next drag a box onto the display area roughly the size and position of the rectangle object required. The exact size and position isn't too important as these are easily modified later.




Note that if snap-to-grid is turned on, the rectangle created will have all of its boundary lines coincident with grid lines, and also the minimum size of the rectangle will be one grid square. See later in this document for more information on grids.

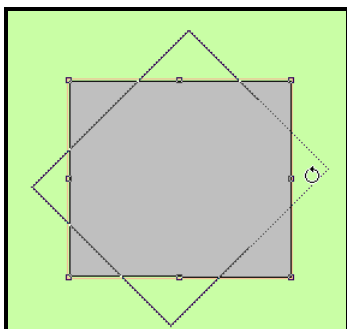
*Holding down the **shift** key while dragging the box causes the box to be square. This is true both while creating the object or adjusting its size later (see next topic).*

How Do You Move Or Resize An Object?

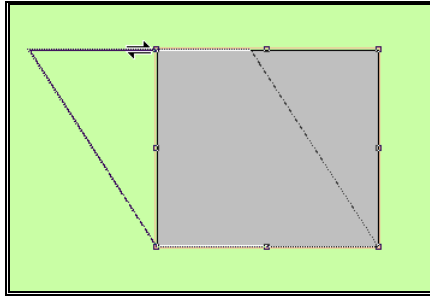
In the example above you can see that the rectangle object has “drag-handles”, the differently coloured blobs at its corners and the mid-points of its edges. Pick the **Selection** tool (see earlier); position the cursor over one of the drag handles and then click-and-drag. The object will be resized. By clicking in the centre of the object and then dragging, the object can be moved.

Orientation Techniques

- **Resizing.**
Click on one of the handles and drag it. If the handle is on a side the object will be stretched in the direction that you drag. If the handle is in a corner the object will be stretched in both its length and its height. If you hold in the shift key whilst dragging a handle the object will scale equally in both axes.
- **Rotating.**
If you hold in the ctrl key while clicking over a corner handle, the cursor will change to show the rotation cursor.  Click and drag from a corner point to rotate the image. Holding down the shift key as well as ctrl will bind the rotation to 15 degree increment.



- **Skewing.**
If you hold in the ctrl key while clicking over an edge handle, the cursor will change to show the skew cursor \Rightarrow or \Downarrow . Click and drag from the edge point to skew the image. Holding down the shift key as well as ctrl will limit the skew to motion in one plane.



The Undo and Redo Commands.

When performing operations on a Display you may make a change that you did not intend. Display Builder maintains a list (currently limited to 100) of operations that have been performed and allows the user to step back through this list **undoing** the operations. Operations that have been **undone** can be **redone**. This technique is used in many applications such as drawing packages or word processors. The undo/redo commands are accessible from the **Edit** menu and the shortcut keys ctrl-Z and ctrl-Y respectively.

Operations that can be undone are:

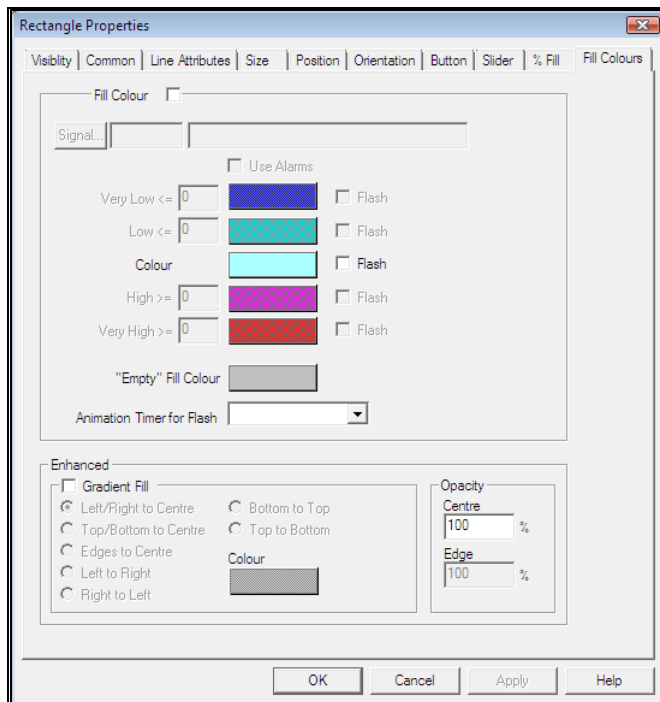
Create Object, Delete Object, Move/Size, Align/Distribute, Z-Order, Group/Ungroup, Roundness(rounded rectangle object), point positions on line/polygon objects, duplicate, load from library and property changes.

Currently, changes to properties of **intelligent objects** cannot be undone. Changes to such things as the background colour, snap-to-grid, grid size etc. cannot be undone.

How Do You Define the Object's Properties?

Access to the properties of an object is via the right-hand mouse button. Select the object that you want to define properties for by left clicking on it. It will indicate that it has been selected by adding the drag-handles mentioned above to its boundary. You can then press the right hand mouse button with the cursor either over the selected object or over the background (i.e. over **no object**). As a shortcut, you can simply place the cursor over the required object and right click, it will automatically be selected. However, right clicking over the background will always bring up the properties for the currently selected object.

Once you have right clicked on the required object, its property sheet will be displayed. Objects have many properties that can be set and these are made available on a **tabbed dialog**. Each tab is specific to one particular property.



To bring a particular property page to the top of the stack, simply left click on its tab.

What Makes Display Builder Objects Dynamic?

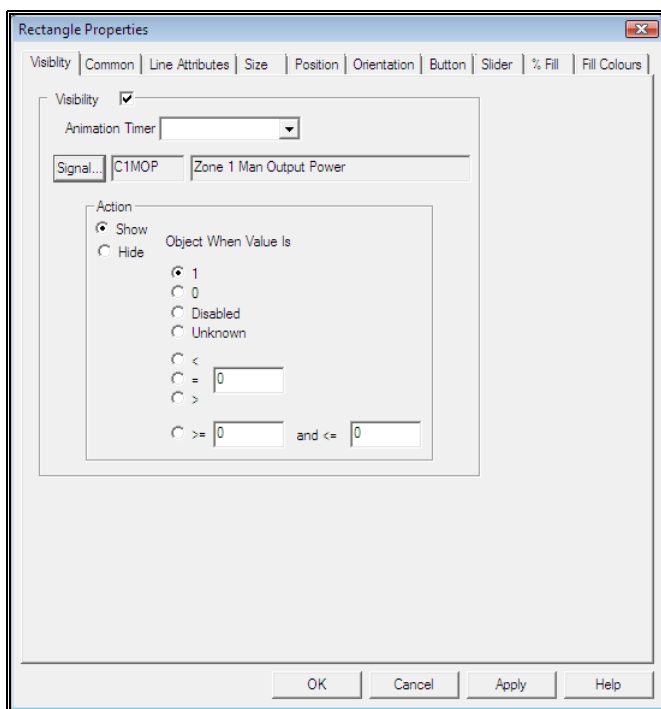
Almost all of the properties of an object can be linked to a signal value. Similarly, the various different properties of a given object can be linked to different signals so that, for instance, one signal may control an objects size while another controls its colour and a third controls its position. By defining suitable dynamic properties for objects realistic, vibrant Display Builder displays can be created that convey information in a clear and precise manner. While most properties are applicable to all types of object, some properties are specific to certain objects.

In addition to being controlled by signals, some object properties can be controlled by internal timers. These are especially useful for producing smooth motion and flashing effects, giving better performance than signal controlled actions which may be subject to queuing effects.

General Object Properties

Let's take a look at each tab on the property sheet for a rectangle object.

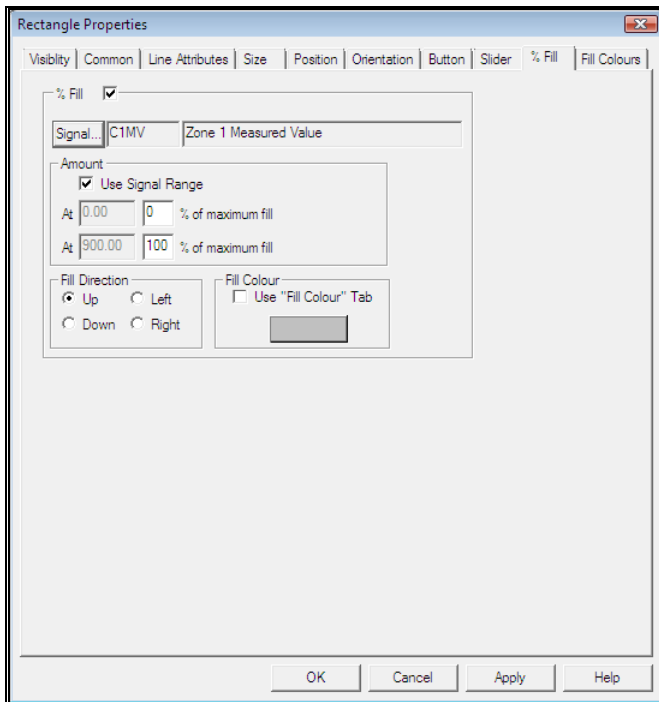
Visibility



The visibility of an object may be determined by the state of a signal or of a timer. This property page (in common with all property pages) allows the governing signal to be defined while further fields allow us to define the signal state that causes the object to become visible or invisible. This can be when the signal value is 1 or 0 (particularly applicable to Digital signals) or when the signal value is >, < or = to a particular value (applicable only to non-digital signals). The signal states **Disabled** and **Unknown** can also be used to cause an object to become visible. A range over which an object can become visible can be created using the >= and <= boxes, also the show and hide radio buttons determine whether a condition shows the object (makes it visible) or hides the object (makes it invisible).

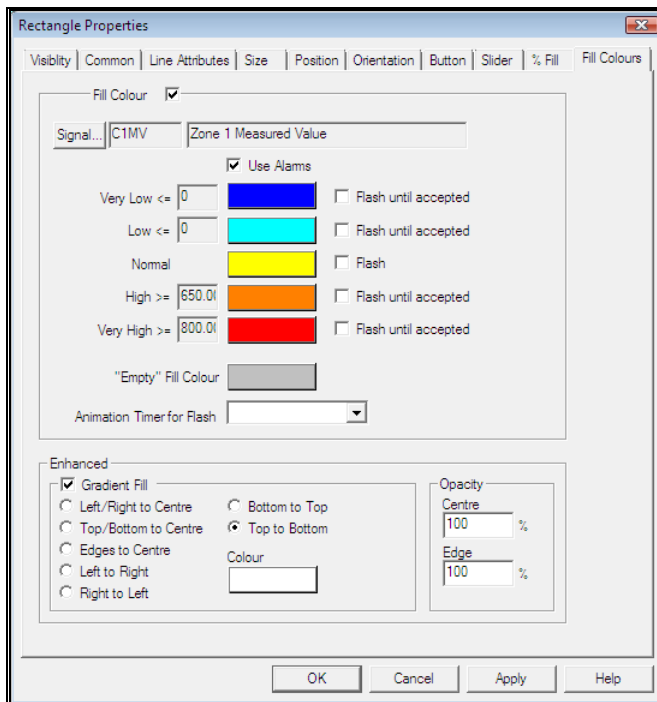
In addition an **Animation Timer** can be specified which will cause the object to become visible or invisible in relation to a built-in timer within Prodigy. See later for more information on **Animation Timers**.

% Fill



An object may be being used to mimic a tank containing liquid. The level of the liquid in the tank can be shown by filling the object in a given colour up to a height that is determined by the ratio of the signal value to the range specified on the form. For instance, if the signal value is 90 and the signal range is 40 to 140, the object will be filled to 50% of its height. The range can be entered manually or the signal range specified in the signal database can be automatically selected. The object can be filled from the left, right, top or bottom. It can be filled in a fixed colour, chosen by clicking on the colour chooser button, or the colour can be made to alter dynamically by using the properties of the “Fill Colour” tab.

Fill Colour



Sometimes the colour of an object may need to change to mimic certain conditions within the plant, such as temperature. For instance blue may be used to indicate cold and red to indicate hot. Prodigy Display Builder allows five colour states per object – one for **normal** and two each for values **above** and **below** normal. These can be mapped to the alarm state of the signal by checking the **Use Alarms** box, otherwise the four trigger points must be entered manually.

When a signal value is processed it is compared against these values (or the alarm state) to decide which colour the object should be. The trigger points are processed as follows:

- If the signal value \leq Very Low use the Very Low Colour.
- If the signal value \leq Low use the Low Colour.
- If the signal value \geq Very High use the Very High Colour.
- If the signal value \geq High use the High Colour.
- Otherwise use the Normal Colour.

If any of the **Flash** boxes are checked then the object will flash if the signal is in the state that the Flash check box relates to. The colour will flash between the **Normal** colour and the colour specified for this value band. Select the timer to be used to control the flashing in the **Animation Timer for Flash** box at the bottom of the dialog. See later for more on Animation Timers.

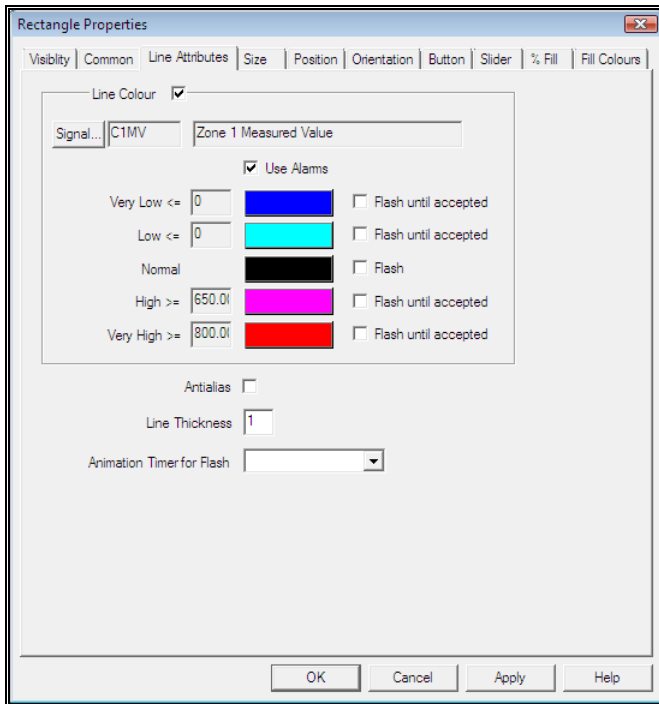
This dialog also allows the **“Empty” Fill Colour** to be set. Looking back at the **% Fill** property we see that it allows us to specify that a portion of the object will be filled with a colour that we specify here. The **“Empty” Fill Colour** is the colour that the remainder of the object will be filled with.

The **Enhanced** grouping of fields allows you to specify some special filling methods. Gradient fills may be selected that specify two colours, the fill blending smoothly from one to the other in a specified direction. The initial colour (the “from” colour) is that specified in **Fill Colour** and the transition (the “to” colour) is that specified in **Enhanced**.

In addition (or alternatively) a degree of opacity may be specified for the fill colour. Two opacity values can be given (where 0 is transparent and 100 is fully opaque), one for the “edge” and one for the “centre”. Edge is actually the “from” colour and centre is the “to” colour, unless no gradient fill is selected, in which case Centre is simply the selected colour.

Be aware that using gradient fills and opacity can have a performance impact, especially if the blocks of colour are individually large.

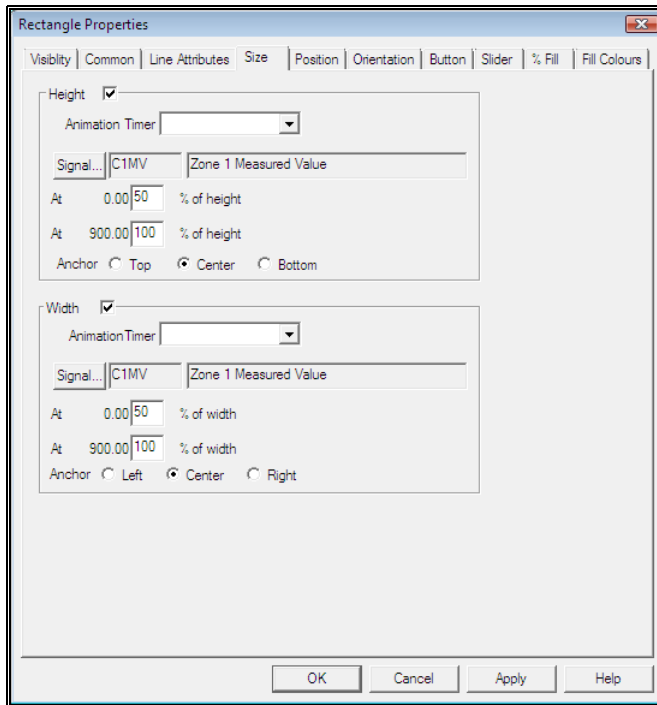
Line Attributes



Lines, whether they be line objects or the outlines of rectangles, ellipses etc., can have dynamically changing colours which operate in the same way as the fill colour described previously. In addition, the thickness of the line can be specified, though this property does not change dynamically.

Checking the **Antialias** box makes the system antialias the lines for the current object, which gives a much smoother look to the object (especially with curves and lines that are at certain gradients) but the trade-off as always is in performance. Antialiased lines are more intensive to draw than standard lines.

Size

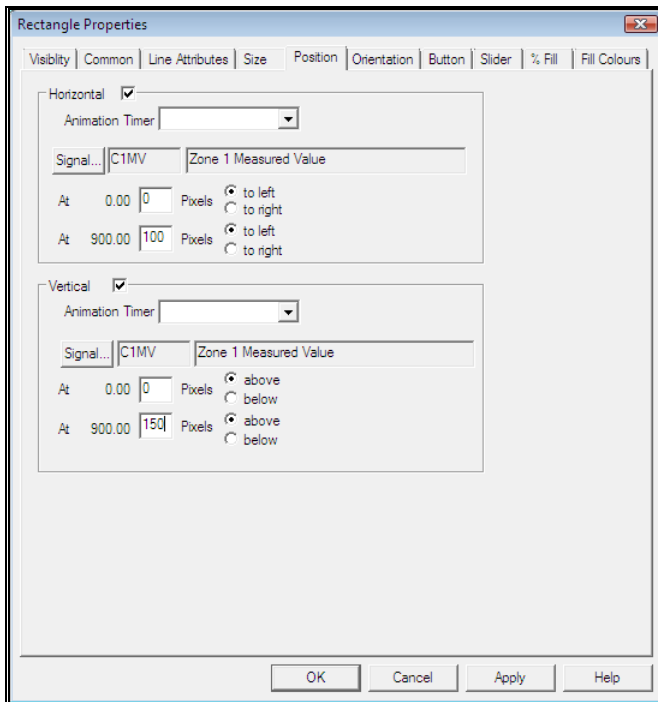


The size of an object can change dynamically upon a change in signal value. The height and width can change independently of each other and can be dependent upon different signals or the same one. The size of the object at each end of the signal range (as specified in the signal database) is expressed as a percentage of its original drawn size. The size of the object is then calculated by interpolating the actual signal value between these two percentages.

The anchor point specifies in which direction the object “grows”. If the width was increasing and the object was anchored “Left”, the object would grow to the right. If it were anchored “Right” it would grow to the left. If it was anchored “Centre” it would grow to the left *and* right. Similarly for up/down.

In addition an **Animation Timer** may be specified to vary the size of the object. If the object size is being varied by a timer then a signal can also be selected that enables the scaling to be turned on or off. Select a digital signal by clicking the **Signal** button. When the digital is **on** the object will resize according to the timer and when the digital is **off** it will stop scaling.

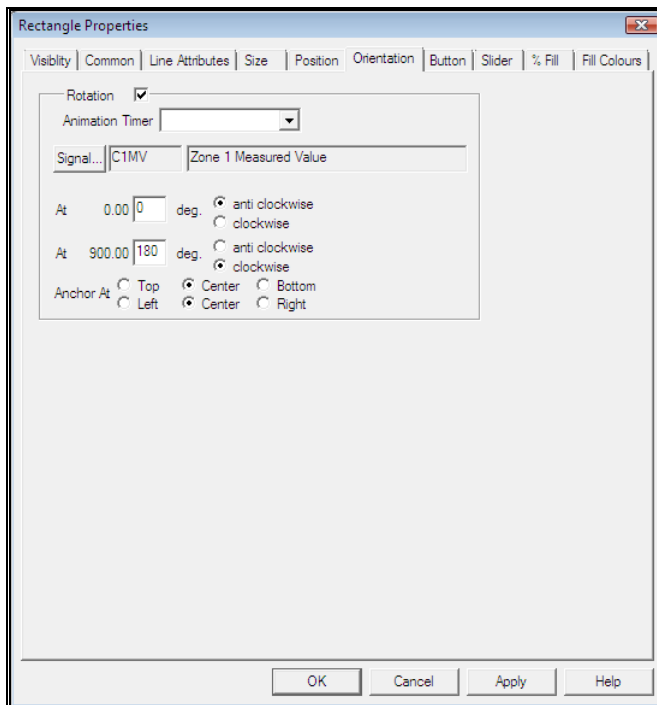
Position



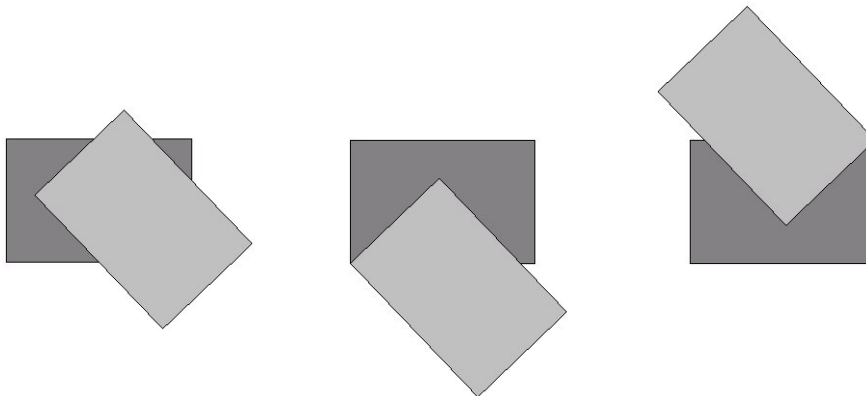
An object can move about the screen, its position depending on a signal value.

This property works in the same way as the **Size** property, only an objects position is determined by interpolating by a number of pixels in relation to its starting location rather than a percentage of its size.

Orientation



An object can rotate on the screen, its orientation depending on a signal value. The orientation of the object is specified at each end of the signal range as a number of degrees clockwise or anticlockwise. The orientation of the object is then calculated by interpolating the actual signal value between these two angles. The anchor point selected determines how the object rotates. Some examples are given below.



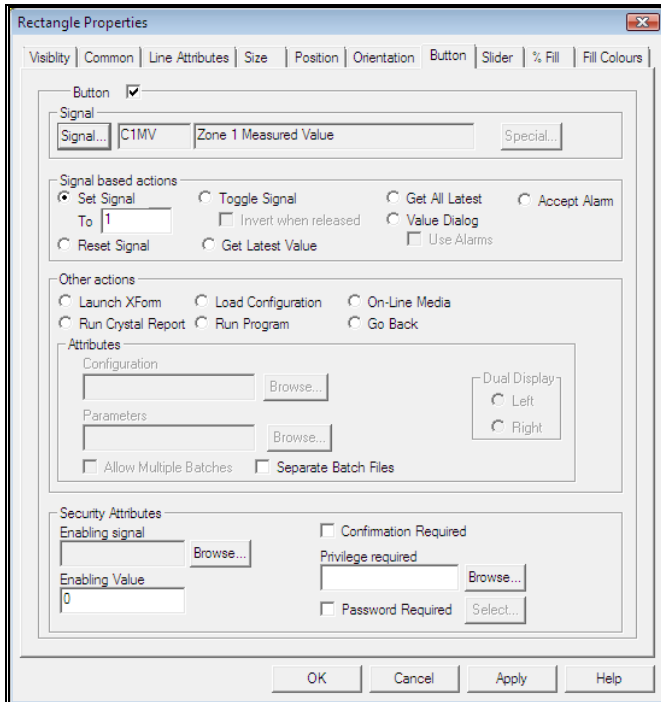
Anchored Bottom Centre

Anchored Bottom Left

Anchored Top Right

An **Animation Timer** can be applied to the orientation in the same way as it can to size and position, the timer controlling orientation and the (digital) signal controlling whether or not any action is taken.

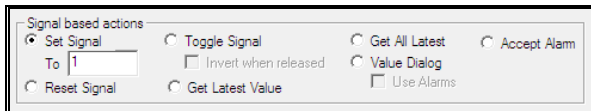
Button



An object can be given “Button” properties, meaning that clicking on the object can cause it to perform some action. Not all of the actions available require a signal to be selected. Some options on the dialog will become available only when a certain action is chosen.

The available actions are:

Signal Based Actions



Set Signal.

The selected signal will have its value set to whatever is specified in the **To** field. Be careful when using digitals that the value is 1 or 0 otherwise the result is indeterminate.

Set Signal can have a different effect if the button object is a **Windows Button**. See later in this section.

Reset Signal.

The selected signal will have its value set to 0 and in addition any software driver attached to the signal (such as a software counter) will be reset. If there is any hardware device attached to the driver controlling the signal (e.g. a hardware counter) an attempt will be made to reset the hardware device.

Toggle Signal.

The selected signal (which must be digital) will have its value set to 1 if it is currently 0 and 0 if it is currently 1. If the box **Invert when released** is checked, the signal will revert to its original state once the button is released.

Get Latest Value.

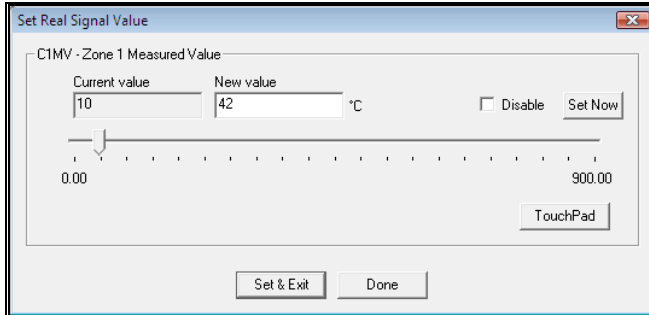
The current value of the selected signal is requested from the signal manager. This is mostly used for signals that have a large update period and will cause the signal manager to request the current value from the plant hardware device.

Get All Latest.

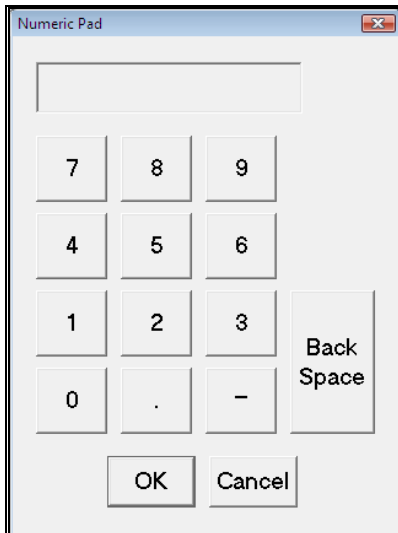
As “Get Latest Value” except that the values of all signals used on the current Display are requested.

Value Dialog.

This causes a dialog to be displayed that allows the operator to enter a new value for a signal. The exact form of the dialog varies with signal type – a digital signal will be displayed as two radio buttons, an analogue signal will have a slider as well as an input field. The dialogs all display the signal name, description and units. The dialog for a real (analogue) value is shown below.

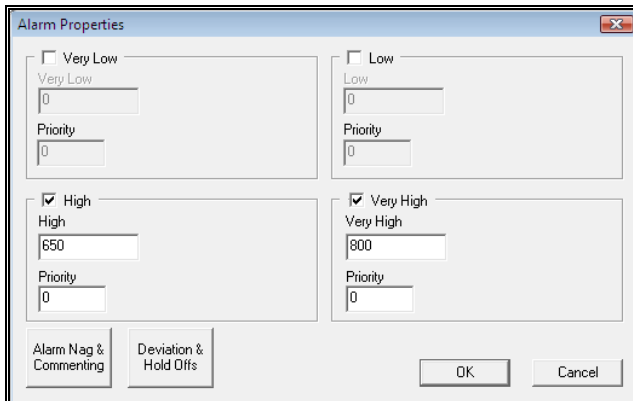


The **TouchPad** button on the Value Dialog displays a large numeric keypad that can be used with touch screen applications.



Use Alarms

When this box is checked, instead of displaying the signal value dialog box (as described above) Prodigy will display the Alarm Properties dialog.



The action of this dialog is as described in **Signal Monitor**, elsewhere in this manual.

Invert when released.

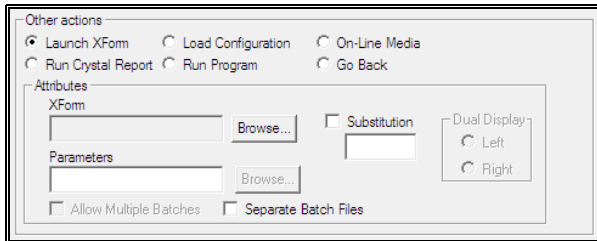
Available only with the **Toggle Signal** option, checking this box will cause the signal value to be set back to either 0 or 1 as appropriate.

Accept Alarm.

If the signal attached to this button is in alarm, clicking the button will accept the alarm.

Other Actions.

Launch XForm.

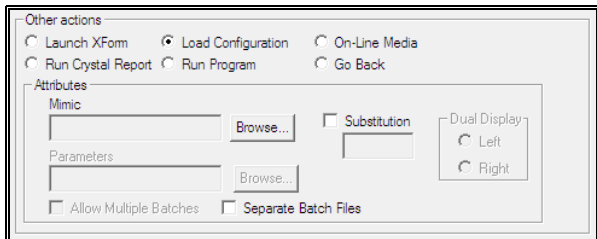


An XForm is a dialog that allows the operator to view and/or set process parameters. These usually relate to the handling of batch data and to allow “recipes” to be selected. The setpoints etc. contained in these recipes can be applied to the signal database and/or written to a batch file. The “**Browse...**” button alongside the **XForm** field allows the required XForm to be selected. When the Display is running and the operator clicks on the button object, the XForm specified here will be displayed.

If the **Substitution** box is checked then tag substitution will be performed on the XForm before it is loaded. Tag substitution works by substituting the leading n characters of each tag name used in the XForm with the characters specified in the substitution field here (n is the length of the substitution string). So, if the string is T2 then the first two characters of each tag name will be replaced by T2. If the box is not checked then the current substitution string will be applied to the XForm (see **Load Configuration**, below). If it is checked and the substitution field is left blank then the XForm is loaded without any substitution taking place.

XForm is described in more detail later.

Load Configuration.



This option allows a different Display Builder display to be loaded when the button object is clicked by the operator. For instance, an overview Display may be constructed showing key information for the whole plant. Clicking on an object on the overview display may cause a more detailed Display of the relevant plant area to be loaded. A button on this detailed Display may take us back to the overview, or to any other area of the plant.

Click the **Browse...** button to select the Display to be loaded.

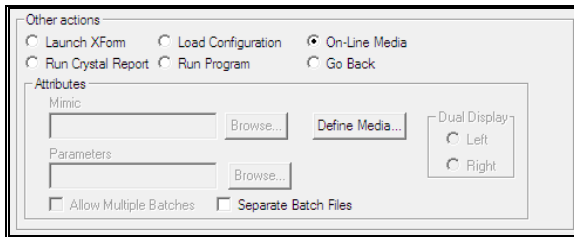
Check the **Substitute** box if tag substitution is to be performed. Enter the substitution characters to be used into the field below the box. Now when the specified Display is loaded all tagnames in the Display will have their leading n characters replaced with the string given here (where n is the length of the string). So you specify T23 as a substitution string all tags in the Display will have their first 3 characters replaced with T23.

If the box is checked but the field is left blank then substitution is effectively cancelled, and the new Display will be displayed with the tags as specified when it was designed.

If the **Substitute** box is not checked and a substitution is already in effect (i.e. a substitution was used when loading the current Display), then the same substitution is passed on into the new Display.

See later in this section (under Making A Better Display) for more information on dynamic tag substitution.

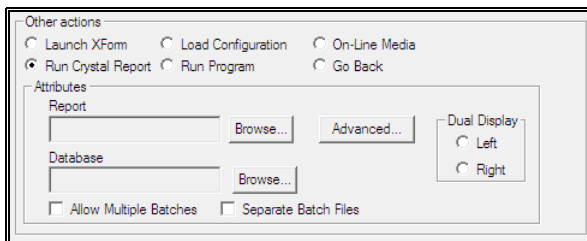
On-Line Media.



Prodigy Display Builder allows information such as maintenance procedures, plant item specification sheets, standard operating procedures etc. to be stored on-line and be displayed when a button object is clicked. The information may be in the form of text, static pictures, moving pictures or sound. A button object may allow the operator to choose from a number of options that are defined here by pressing the “**Define**” button.

On-Line Media is described in more detail later.

Run Report.

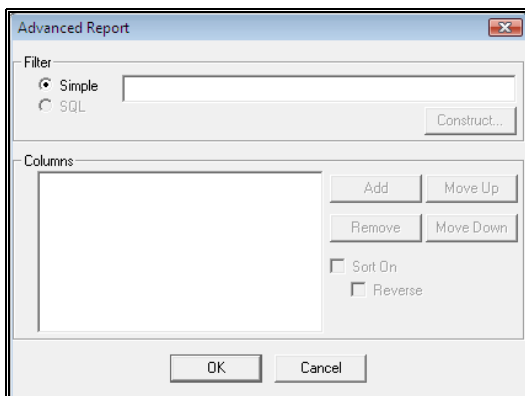


This option means that pressing the button object will cause a batch-style report to be run. The **Report** and the **Database** on which it operates are selected here by clicking the relevant **Browse...** buttons.

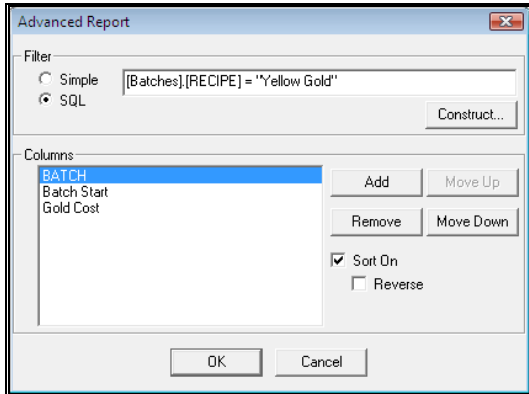
If the **Allow Multiple Batches** box is checked then the **Batch Selection Dialog** that will be displayed when the button object is pressed (i.e. when the **Display** is actually running) will allow multiple batches to be selected.

By default the **Batch Selector Dialog** that is shown when the button is pressed will display the **Batch Name**, the **Batch Start Date** and the **Batch End Date**. It will also show **all** batches. You may require something other than this and so the behaviour may be modified by pressing the **Advanced...** button.

The following dialog will be displayed...



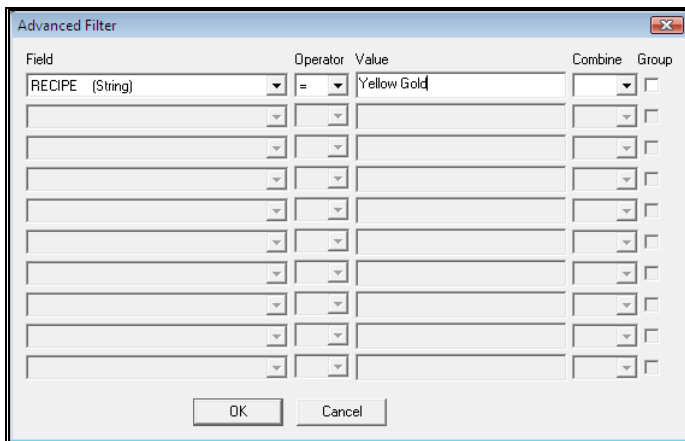
This dialog has a number of fields that are not available. To make them available you must first have selected a **Database** in the previous dialog. If a database is selected, all of the options on this dialog will become available.



The **Filter** option allows you to specify some kind of filter on the records that will be displayed in the **Batch Chooser** dialog. A **Simple** filter will simply be a string with some wildcard characters in it, such as **M61***, which will select all records with a batch name beginning with **M61**.

An **SQL** filter is more complicated and can be used to select records which match any one (or all) of a number of criteria and these criteria can be based on any field in the database, not just the batch name.

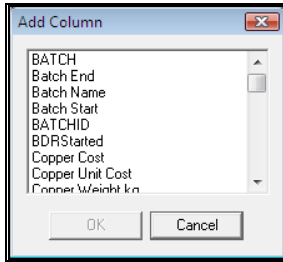
You can type the SQL string in directly (if you are familiar with SQL) or you can use the **Construct...** button to select the **Advanced Filter** dialog.



The drop-down lists at the left of the dialog contain all of the **fields** in the database – select the required field from this list. Next, select the required relational **operator** from the drop-down list to the right of this and then enter a **value** into the text box to the right of that. Note that these should just be entered as plain text, the system will automatically place quotes (") around strings and hashes (#) around dates, as required by SQL. If you require extra criteria select either **AND** or **OR** from the **Combine** drop-down list. Finally, the **Group** check-box is used to group criteria together, in effect it **brackets** them. It works as follows: all items that follow a group check-box (including the one on which the group box appears), up to the item preceding the next group check-box are bracketed together.

The SQL statement shown here will select all records for which the Recipe that was chosen was “Yellow Gold”.

Back on the **Advanced Report** dialog, there is a group labelled **Columns**. This allows you to choose which columns appear on the batch selection dialog. Each column relates to a database field, so clicking the **Add** button displays a list of all of the fields in the batch file.

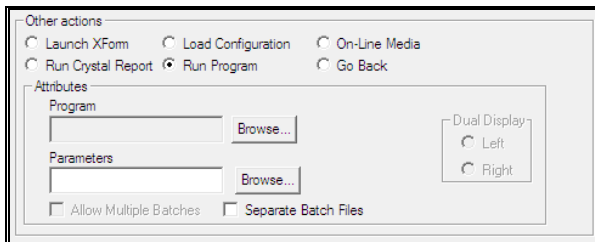


Select the required field. Use the **Move Up** and **Move Down** buttons to arrange the order of the fields, which will be the order of the columns on the dialog. Check the **Remove** button to remove any unwanted fields. Select the **Sort On** and (optionally) the **Reverse** buttons to determine the initial sort state of the records within the Batch Selection Dialog when it is displayed.

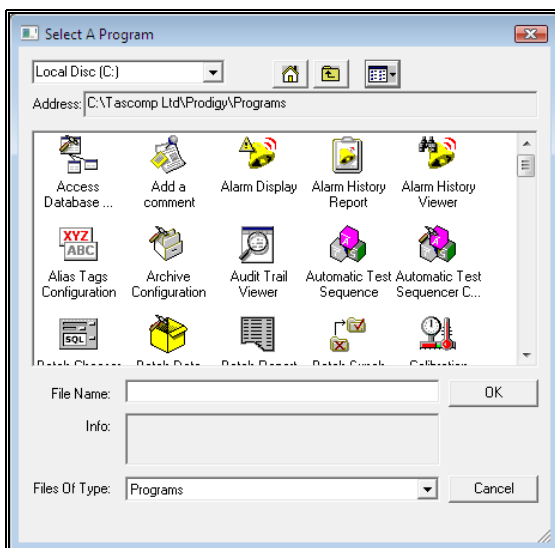
When the report is run and the **Batch Chooser** dialog is displayed, the fields chosen here will be displayed in columns. Each column may be sorted in ascending or descending order by clicking on its column header. The records that are shown will only be those that match either the **Simple** or **SQL** filter that was selected.

*The Run Report facility requires the installation of Seagate Crystal Reports, supplied as part of the **Prodigy Complete** version.*

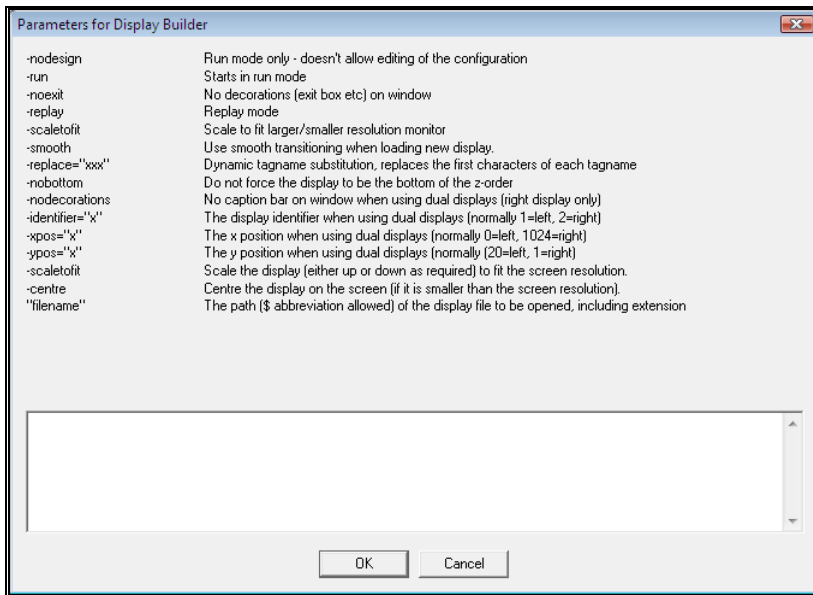
Run Program.



This allows any executable to be run when the button is pressed. Use the **Browse...** button to select the required program. The Prodigy Program Selector (below) will be displayed. This is explained in more detail in the section about Menu Bar. For programs that are part of Prodigy extra information about the program is displayed in the **Info** field of the dialog. Programs other than those that are part of Prodigy may be selected by browsing to their location and selecting them.



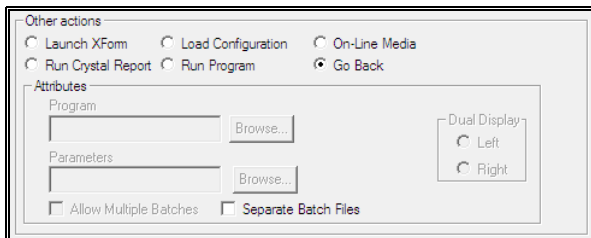
Enter any **Parameters** that the program requires in the relevant box. For Prodigy programs the **Browse** button next to the Parameters field may be used to select the parameters help dialog. Fill any required parameters into the area at the bottom of this dialog and press **OK** when ready.



*If no program is specified there are a number of **special** parameters that can be entered.*

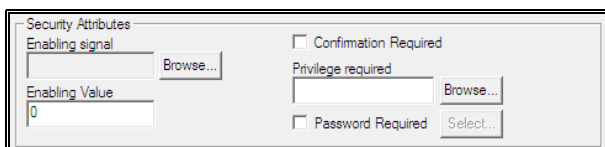
*Specifying the parameter **-print** means that when the button is pressed the **Display** will be printed. This allows you to provide a visible alternative to using the **ctrlP** key to perform a print.*

Go Back



At run-time, Display Builder maintains a stack of which configurations have been loaded. The **Go Back** facility allows you to place a button on a display that will take you back to the last Display that was displayed. This means that a given Display can be loaded from any number of other Display Builder displays, and you can always return to the display that invoked it with a “Back” button. You could not achieve this by using the “load configuration” method. The stack is reset if you go into editing mode, or if a new Display is selected from the menu bar, or via a SLANG call. The stack is only maintained for Displays called up via buttons actually on the Display Builder display.

Security Attributes



Enabling Signal & Enabling Value.

This is a signal that is checked to see if the button can be pressed. If the value of the **Enabling Signal** is equal to the **Enabling Value** as entered in the box above then the button can be pressed, otherwise it can not.

*This section is only available when the object is not a “real” button – in that instance the enabling functions are specified within the **Special...** section (see later).*

Confirmation Required.

When setting up a button that will set, reset or toggle a signal, the designer can specify that a confirmation must be issued. This takes the form of a dialog stating the description (or name) of the signal and its proposed new state. The operator must confirm that this is the required action before the signal value will be changed.

Privilege required.

Specifying that a privilege is required in order to access this button can limit the buttons availability to a certain group of operators who have been granted that privilege. Use the **Browse...** button to display a list of available privileges. Extra privilege types can be created within the Users program so that different categories of operators can be allowed access to different buttons on the same Display.

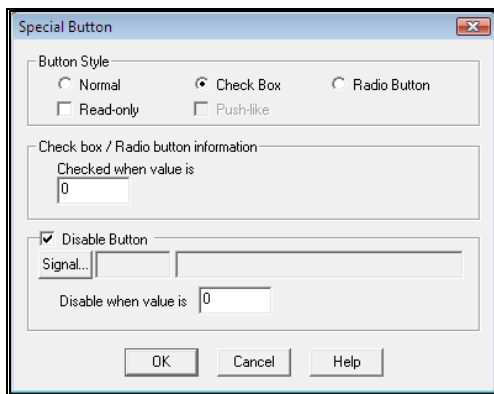
This facility will most likely be used in place of the Password required option in systems where Prodigy security is implemented, though the two can be used in conjunction.

Password Required.

Specifying that a password is required can protect a button. If this option is checked the user should also specify the required password by pressing the "**Password...**" button. When the Display is running, pressing the button will present a dialog requesting the password and the button will have no effect until this password is given. When Prodigy security is implemented it is more likely that Privileges will be used rather than passwords to protect buttons, though the two can be used in conjunction.

Special Button Properties

The **Special...** button on the Button Properties tab is only available when the object whose properties are being defined is a **Windows Button**. The **Special Button Properties** dialog allows the user to configure how the button appears and behaves.



The properties are:

- **Button Style**
 - **Normal.**

The button has a normal appearance. Clicking the button will cause the action defined on the Button Properties tab to be performed.
 - **Check Box.**

The button has the appearance of a square with a label to the right of it (in the dialog shown above Read-only is a check box). This box has a tick in it when the value of the controlling signal for the button is equal to the value entered in the "Checked when value is" field. In addition, if the button action chosen on the Button Properties tab is "Set Signal", the signal value will be set to the value given in the "Checked when value is" field.
 - **Radio Button.**

The button has the appearance of a circle with a label to the right of it (in the dialog shown above the Normal, Check Box and Radio Button fields are radio buttons). This button has a black centre to it when the value of the controlling signal for the button is equal to the value entered in the "Checked when value is" field. In addition, if the button action chosen on the Button Properties tab is "Set Signal", the signal value will be set to the value given in the "Checked when value is" field.

Radio buttons and check boxes can be used to show and/or set the value of finite state or digital signals. If a finite state signal goes from 0 to 5 then place six radio buttons on the Display; make the first be checked when the signal value is 0; the second be checked when the signal value is 1 and so on for all 6 states. As the signal value changes a different button will become checked. Pressing one of the buttons will set the signal value to the state that that button represents.

When a check box or radio button is used its label can be automatically picked up from the event text entry for the signal that is controlling it. Set the "Value Display" tab for the object to have the same controlling signal. If you then select **Text**, the event text for the relevant state will be shown. If you choose **Value**, the numeric value that that state represents is shown.

When configuring radio and check box sequences it makes a great deal of sense to set up one button first and then duplicate it the relevant number of times. This way only the "Checked when value is" field needs to be modified.

- **Push-like.**

This option is only available with Radio buttons. The button will have the appearance of a normal button when it is not "checked" but when it is "checked" it will take up and indented appearance.

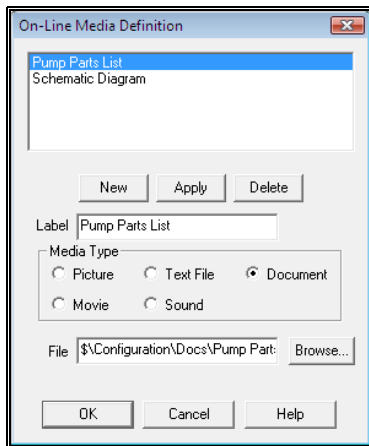
- **Read-only.**

This does not affect the appearance of a button but does prevent it from causing any actions when it is pressed. It is used so that a range of check box or radio buttons may be used to indicate a process state without being able to affect that state.

- **Disable Button.**

This allows the button to become *disabled*, which means that it has a greyed out appearance and pressing it will have no effect. When the button will be disabled is determined by the value of a digital signal that is selected here. The value of the signal that will cause the button to become disabled (0 or 1) must also be entered.

On-line Media



To create on-line media options for a button check the **On-Line Media** radio button on the property sheet and then press the **Define...** button. This will display the dialog shown above.

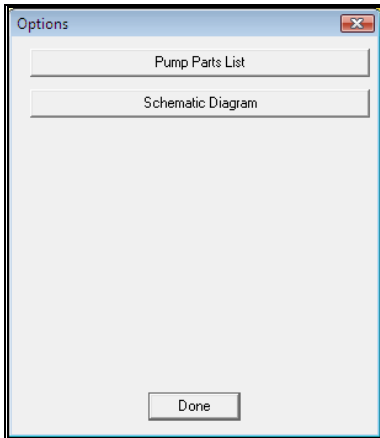
To create an entry in the media definition table press the “**New**” button. This creates a dummy entry called “New”. Change the label to some meaningful text and select the relevant media type.

- **Picture.**
This is a static bitmap image that may have been stored in one of three common bitmap formats, **BMP, JPG** or **PCX**.
- **Movie.**
This allows you to pick either a **.AVI** or a **.MPG** file, both of which can contain moving pictures and sound. The file is replayed through a standard windows control for displaying movie files.
- **Text File.**
This displays a plain text file of the type created by the Notepad program or any other plain text editor. The extension that is expected for this type of file is **.TXT**. No special formatting is recognised.
- **Sound.**
This allows sounds to be played back via the PC’s soundcard and speakers. The expected file extension for this type of file is **.WAV**.
- **Document.**
This allows a **Microsoft Word™** or **Adobe Acrobat™** document to be viewed. The expected file type is **.DOC** or **.PDF** respectively, although the Word viewer can view any type of file recognised by Word (e.g. **.RTF, .TXT**).

The File to be used is entered in the relevant field or the “**Browse...**” button can be used to select the required file. The Browse button will display a file chooser dialog that will allow the user to select files of the correct type for the specified media type, i.e. files with the correct extension, as detailed above.

When you have finished specifying this media entry press “**Apply**” to commit the changes. Create any other media definitions that may be required for the button action being defined. When all media items have been added, press “**OK**” to apply these changes to the button action. Pressing “**Cancel**” means that all of the changes to media definitions that have been made here will be lost.

When Display Builder is in Run mode the operator may press the button that you have just defined. What happens next depends on how many media items were defined. If more than one has been defined a dialog is displayed with all of the options on it.



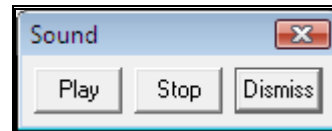
If there is only one option, the relevant file viewer is called up to display the information specified.

File Viewers

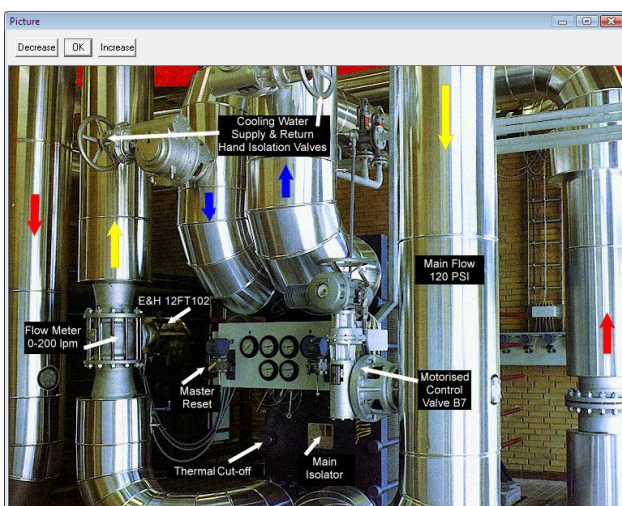
The operator will select the desired option from the list and a further dialog will display the Text file, Picture, Audio-visual image, sound file or formatted document.



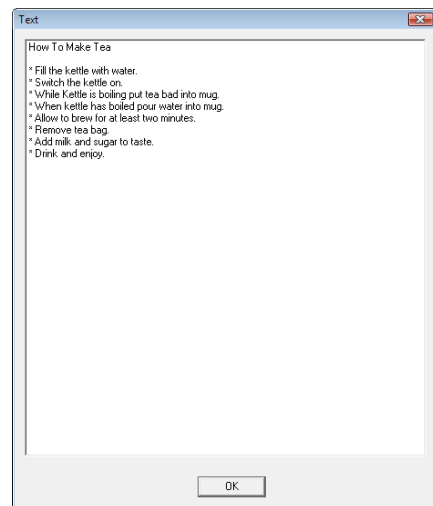
AVI/MPEG File Viewer



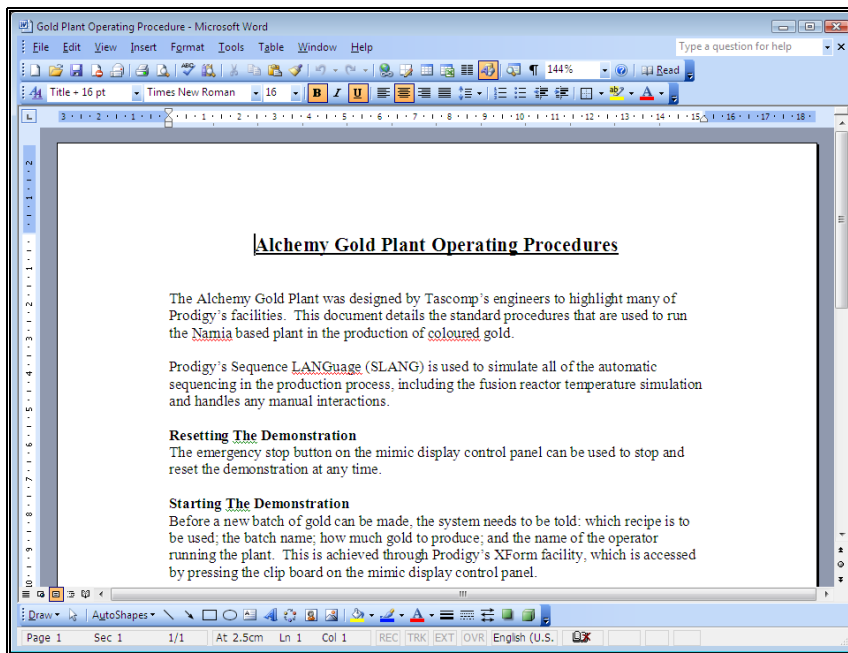
WAV File Player



Picture file (JPG, BMP, PCX) Viewer



TXT File Viewer



Word Viewer

The AVI and WAV file dialogs have “**Play**” and “**Stop**” buttons. When the dialog appears the picture or sound will be playing automatically. Pressing Stop will stop the image/sound and pressing Play will resume playback from this point. When the image/sound has played to completion the Play button will repeat playback.

The TXT File Viewer will have a scroll bar down the right hand side if there is more text than can fit into the pane.

The Picture File Viewer has buttons to increase and decrease the image size. The image is initially loaded at the size at which the image was saved (unless this is too big for the screen). The “**Increase**” button will double the image area and the “**Decrease**” button will half the image area (i.e. each side will be scaled by a factor of $\sqrt{2}$).

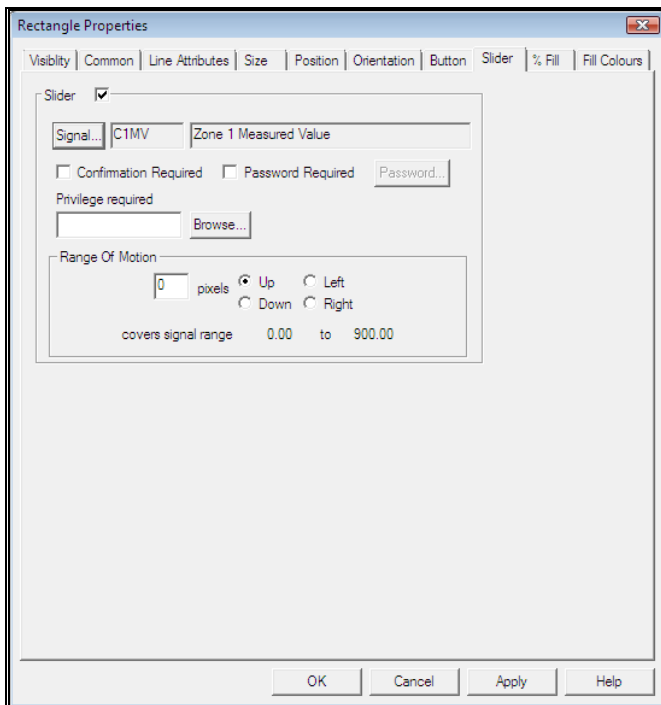
If the chosen item is a **Word** or **Acrobat** document, the system will look for the correct 3rd party viewer associated with this type of document. If you do not have Word or Acrobat installed on your computer, there is a Word viewer and an Acrobat reader on the Prodigy CD. These may be installed free of charge onto any PC, subject to the standard licensing agreement that comes with each product (this may be viewed as part of the product's installation procedure).

When Prodigy finds the relevant viewer for the type of document selected, it places it's path into it's own registry key and then launches the application. If the application that has been selected is the wrong one for that type of document, your supplier, integrator or Tascomp can tell you how to modify this registry entry so that the correct file viewer is used.

If no viewer can be found automatically, Prodigy will display a browser dialog that allows you to locate the required file viewer.

*It is suggested that the developer go through this procedure as part of the development/commissioning cycle. It is **not** something that should be left to the operator!*

Slider



Sliding an object across the screen may be used as a method of setting a signal value. Left clicking on the object and then dragging it in the direction specified on the property page performs this operation.

The object is allowed to move only within the range specified on the property page. The signal value is calculated from the percentage of the range that the object has been moved, applied to the range of the signal. This operation is most often used when the object will be part of a composite object that forms a slider gauge.

*As with **Button Properties**, when setting up a slider the designer can specify that a confirmation must be issued, a password must be given or a privilege is required. See the relevant section of **Button Properties** for more information on these features.*

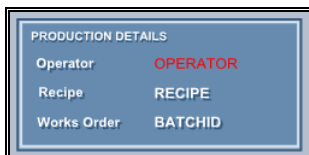
More Specialised Properties

Some object types have only a limited number of properties available to them, for instance a Text object. These properties are described here.

Grid

A grid object is simply a box filled with vertical and/or horizontal lines. It provides a graticule that can be placed over an area of the display to improve the look of it or to aid with read-off.

No Grid:



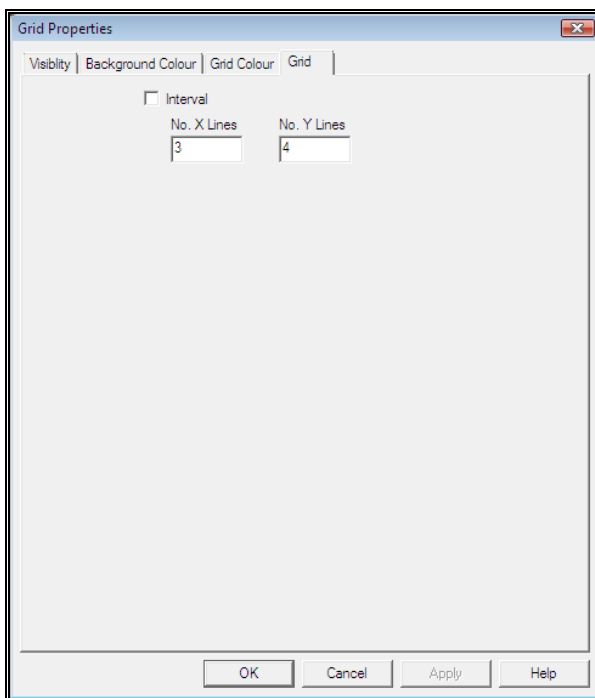
Basic Grid:

PRODUCTION DETAILS	
Operator	OPERATOR
Recipe	RECIPE
Works Order	BATCHID

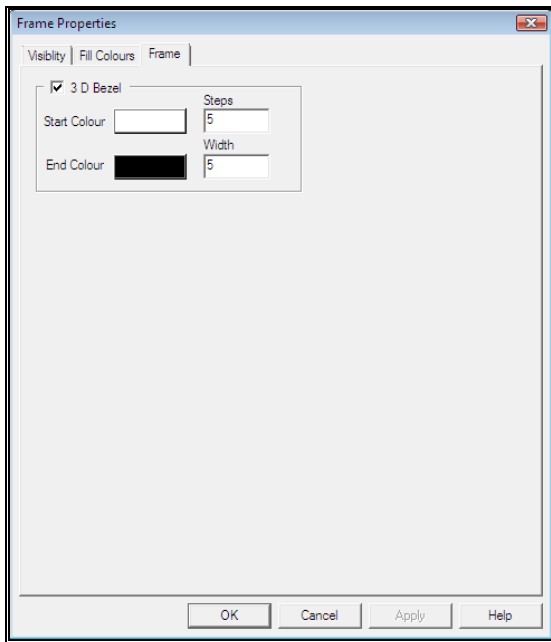
Grid with background colours:

PRODUCTION DETAILS	
Operator	OPERATOR
Recipe	RECIPE
Works Order	BATCHID

The Grid Properties dialog is shown below. You can set the number of lines to show in the X and Y directions (including the edges) or specify the interval between the lines. In this latter case, as the object is dragged bigger or smaller more or less lines will be drawn in the grid.



Frame

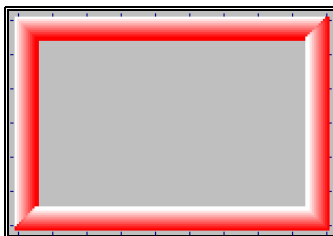


Select the start and end colours by clicking the colour chooser buttons. The start colour is the top and left outer edges of the frame and the bottom and left inner edges. The end colour is the bottom and right outer edges and the top and left inner edges.

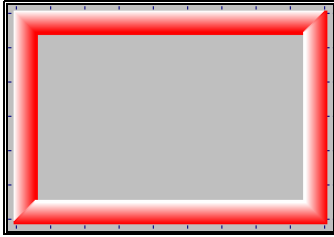
The Width field dictates how wide the borders of the frame are and the Steps field dictates how smooth the transition is from the start colour to the end colour. The higher the values of each of these figures, the more processing the frame takes to draw, so choose carefully. Choosing a number of steps that is larger than the width has no effect, the steps should always be less than or equal to the width. In general, a number of steps set to roughly half the width gives acceptable results.



Width 20, Steps 5

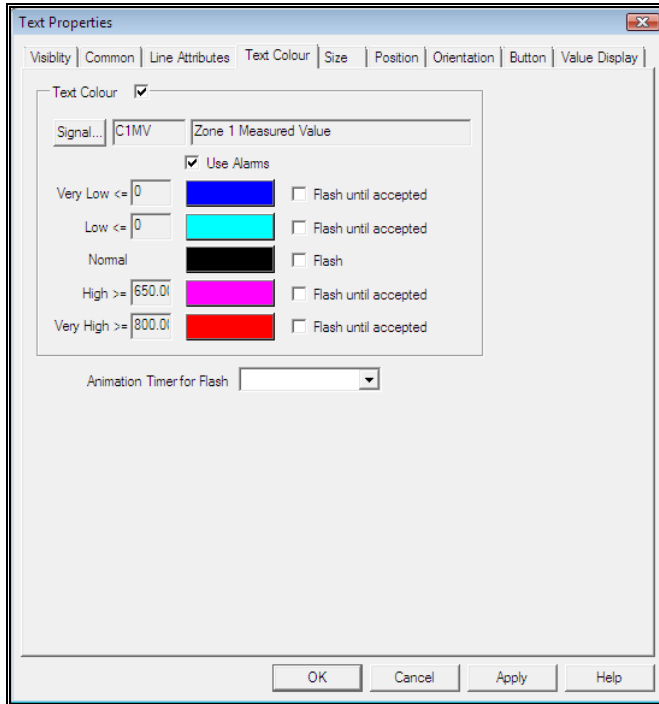


Width 20, Steps 10



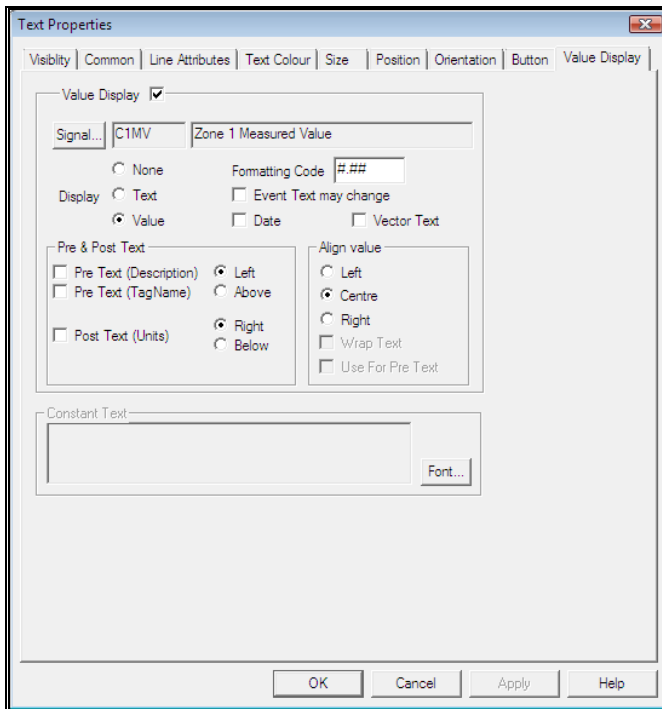
Width 20, Steps 20

Text Colour



This property page controls the colour of a text object. Its function is exactly as for the **Fill Colour** and **Line Colour** property pages described previously.

Value Display



This property page allows a text object to display the value of a signal. The value of the signal can be displayed in two different ways. Its numeric value can be shown or, if the type of the signal is **digital** or **finite state**, its event text entry can be displayed.

*Event Text can be specified for digital signals by using the **Signal Database Configuration** program. This allows an item of text to be specified for each state of the digital signal (0 or 1) to provide a more meaningful feedback. For instance, 0 may mean “Off”, “Stopped”, “Out”, “Closed” etc. and 1 may mean “On”, “Running”, “In”, “Open” etc. The text object may display this text. Finite state signals are similar except that their range is not limited to 0 and 1.*

The Display options are as follows:

- **None.**
Do not show anything of the value of the signal. The intention must be simply to create a label from the name and/or description of the signal.
- **Text.**
For Digital & Finite State signals only. Displays the event text entry associated with the current value of the signal.
- **Event Text May Change.**
If you choose to display the signal value as text the further option **Event Text May Change** becomes available. Check this box if the event text entry may change without the actual signal value changing. This can be made to happen by XForms or by control software and if this box is not checked the change to the event text will not be reflected on the Display until the signal value changes – which may be never!
- **Value.**
Displays the numeric value of the signal.
- **Formatting Code.**
If you choose to display the signal value as a number, a formatting code may be given. This consists of a number of hashes (#) followed optionally by a dot (.) and some more hashes. The first set of hashes control the *minimum* number characters that will appear before the decimal point and the second set of hashes controls how many decimal places are shown. If the first character of the formatting string is a 0 then the output will have leading zeroes instead of spaces. If no decimal and second set of hashes is given, the number will be displayed as an integer.
Examples: if the value is 12.345 then

gives 12.34
 ###.# gives ∇12.3 (where ∇ represents a space)
 0##.# gives 012.3
 ## gives 12.

- **Date.**
 Check this box if the numeric value of the signal is to represent either a date (e.g. 30-Apr-1995) or a time period (e.g. 3d 2h 24m). Specify a suitable formatting code for the type of output that you want to see.
- **Formatting Code.**
 To display the value as a date/time use the following format codes DD:MM:YY hh:mm:ss or DD:MMM:YYYY hh:mm:ss. This shows the day, month, year, hours, minutes and seconds respectively. In the case of MM it shows a two-digit number for the date, for MMM it shows a three-letter abbreviation of the month. YY shows a two-digit year, and YYYY shows a full four-digit year.
 Not all codes need to be specified, for instance to just show the current month, date and time use DD:MM hh:mm:ss. Any *sensible* combination is allowed, so long as the field codes are consecutive (e.g. DD:YY is **not** allowed).

 To display the value as a *time period*, just use the hh:mm:ss part of the formatting code. The value will then be interpreted as just the elapsed number of hours, minutes and seconds. This can be used to display the value of **state time** counters for instance. If your counter runs to more than 1 day (more than 24 hours) the displayed value continues to count up and up. 10 days would be displayed as 240:00:00 (240 hours). There is no way to make it show days as days.
- **Pre & Post Text.**
 Optionally, automatic pre and post text can be added to the field. Pre-text can be either the signal tag name, its description or both, and post text is the units. These are all specified in the signal database. This makes it easier to construct fields of the type <Signal Description> <Signal Value> <Signal Units>, which may appear as
 Furnace Temperature: 3200 Deg. C.
- **Alignment.**
 The value may be made to align to the **left**, **centre** or **right** of its display box. It is usual to align numeric values to the right and text values to the left, though this is entirely at your discretion. For the example shown above, left, centre and right alignment would give
 Furnace Temperature:3200 Deg. C.
 Furnace Temperature: 3200 Deg. C.
 Furnace Temperature: 3200 Deg. C.
- **Use For Pre-Text.**
 The alignment works in slightly different ways when **None** is selected as the display type and **Use For Pre-Text** is checked. In that instance the output for left, centre and right would be
 Furnace Temperature: Deg. C.
 Furnace Temperature: Deg. C.
 Furnace Temperature: Deg. C.
- **Wrap Text.**
 If this option is **not** checked, all output is on one line and may be clipped if the bounding box is not wide enough. Checking the box means that text will wrap to the width of the box, lines being broken at spaces initially.
- **Constant Text.**
 If the **Value Display** box is not checked then the text and/or value to be displayed is not taken from a signal but rather from the contents of this box. The **Alignment** and **Wrapping** options described above are available for Constant Text.
- **Font.**
 Use this button to select the font to be used for this text. Not that all text for this item will be in the same font, there is no way to have pre-text, value and post-text in different fonts.
- **Vector Text.**
 Checking this box causes Prodigy to use vector text for this option rather than raster text. This gives more flexibility at the expense of being slightly more “fiddley” to deal with. If Vector Text is selected the text can be rotated and/or skewed, like any other Prodigy drawing object. Stretching it

will also cause it to thicken; line attributes can be applied to the text outline and colour and fill attributes applied to its interior. See the examples below, all based on the initial string “This Is A Test”.

Raster text



Stretched Vector Text



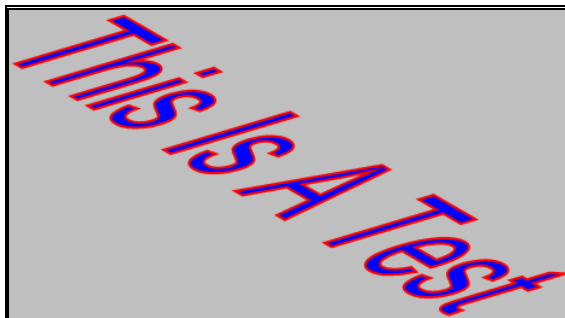
Skewed



With Line Attributes

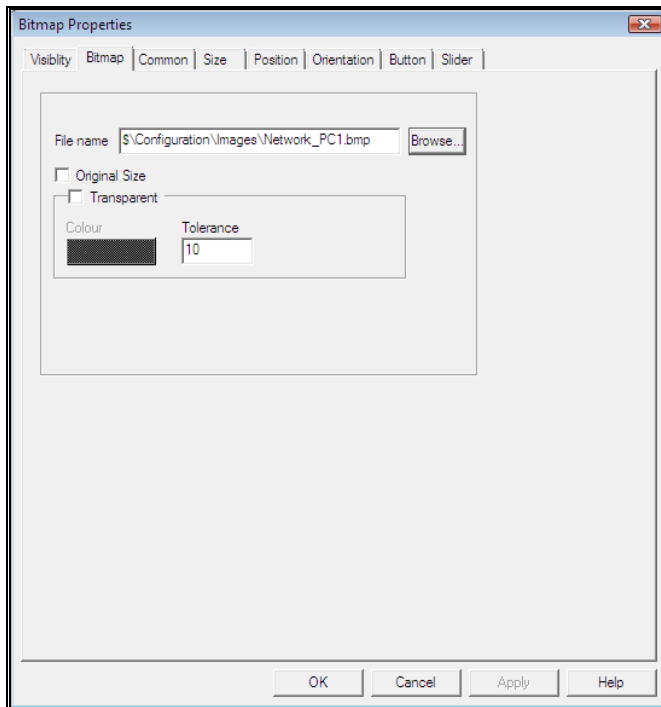


Rotated and with Text Colour Attributes



You can change the text and the font even after any number of stretching, rotating, Skewing etc. operations have been performed.

Bitmap



Bitmaps may be used on Display Builder displays and in common with other objects they have attributes affecting their visibility, size, position and orientation and may also have slider and button attributes. The bitmap attributes property page allows the designer to specify the name of the bitmap to be used and whether or not it is to be transparent.

Bitmaps can be a very useful object as they allow images produced by drawing, painting or rendering packages or even scanned images to be used in a Display Builder display. This can greatly enhance the visual aspects of the Display. However, bitmap images have a drawback in that they cannot change colour or “fill”. Effects can be created though, by having say two bitmaps representing an on and off state and toggling which bitmap is visible by setting its visibility attribute. Similarly, overlaying closed polygons over a bitmap image to provide a “fill” effect, for instance for a tank, can create a good visual effect.

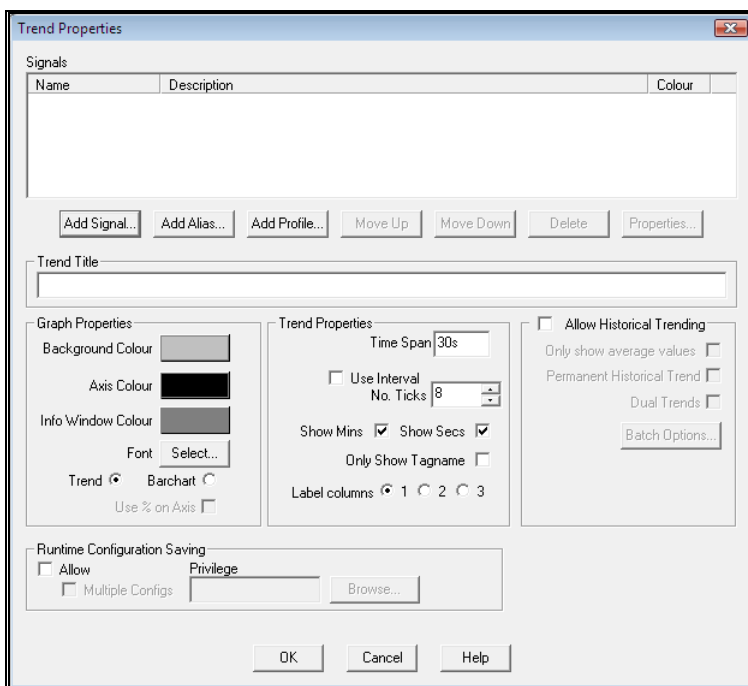
- **File Name.**
This is the name of the file to be loaded into the bitmap image. Use the **Browse** button to select the required file. The types of image file that can be loaded are bmp, jpg/jpeg, gif, tif/tiff, emf, wmf and png. **Note** that due to technical limitations wmf and emf files are currently loaded upside down.
- **Original Size.**
Check this box to display the image at its original size. This is an instantaneous effect and causes the image to resize when you hit the OK button and then the property is removed, so that you can subsequently stretch and scale the image. Applying the Original Size attribute again later allows you to restore the image’s original size.
- **Transparent.**
Checking this option allows you to select a colour in the image that will be treated as being transparent. Select the colour with the colour picker box and set a tolerance limit. This determines how close to the selected colour the colour in the bitmap must be. A tolerance of 0 means the colour must match exactly; a value of 100 means that any colour on the bitmap will match (i.e. a value of 100 is useless and should not be used).
Note that currently bitmaps cannot be rotated or skewed.

Trends & Trend Properties

A Trend object allows real-time and historical signal values to be plotted as graphs. The Trend properties dialog allows any number of signals to be added to the Trend, though obviously the more there are the less readable the data will become.

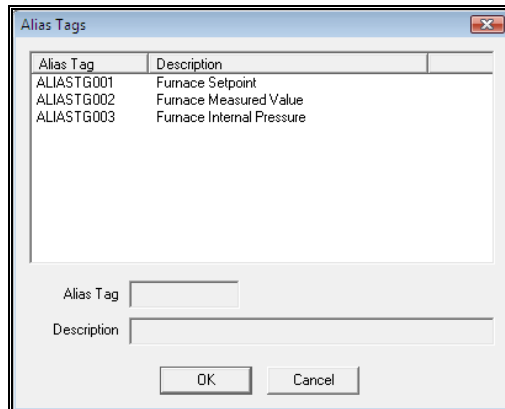
In addition to actual signals, **Alias Tags** may also be added (see the relevant section earlier in this manual for how to define Alias Tags). Alias Tags allow different signals to be shown on a trend depending on which batch is selected. **BDR** must be configured to write into a batch file which **real** tag relates to a given **alias** tag (see **BDR Configuration** for more details). When you configure a trend to use alias tags, nothing is seen of the trend until you enter Historical mode and select a batch. The system then interrogates the batch file to see which real signal relates to the alias tags used in the trend and uses those signals. See the section on defining Alias Tags for a fuller description on how alias tags work.

Finally, a number of **Profile** (or “Ideal”) curves may be added to a Trend and these will be displayed when the Trend is viewed in historical mode. The actual curve may then be compared against the ideal curve. See the relevant section for more on Profile Curves.



- **Add Signal.**
Press this button to add signals to a Trend. The SigScan dialog will be displayed to allow you to choose the required signal. Any signal may be added to a real-time Trend but note that if you choose to switch to a historical Trend only those signals with recording strategies will be plotted.

- **Add Alias.**
Press this button to add Alias Tags to a Trend. The Alias Tags dialog will be displayed to allow you to choose the required tag. Alias Tags can only be seen on historical trends. The Alias Tags dialog is shown below.



- **Add Profile.**
Press this button to add profile curves to a trend. A file chooser dialog will be displayed allowing the required profile definition to be selected.
- **Move Up.**
This button moves the selected trend up in the list.
- **Move Down.**
This button moves the selected trend down in the list.
The order in the list controls the order that the relevant y-axes are drawn on the trend.
- **Properties.**
Clicking this button opens the relevant properties dialog, depending on whether the selected item is a signal, alias or profile. Alternatively you can double-click on the list item to achieve the same effect.
*The colour of the graph for newly added signals is automatically selected from the list **Red, Green, Blue, Cyan, Magenta, Black, Yellow**, depending on which colours have already been used. If all of these colours have already been used in this Trend then **Black** will be selected. You alter the colour in the Properties dialog.*
- **Delete.**
Signals, Aliases or Profiles may be deleted from the Trend by selecting the required item in the scrolling list and pressing this button.
- **Trend Title.**
The title of the trend, drawn centred at the top of the trend object.
- **Background Colour.**
The background colour of the Trend object, over which all axes, graphs, titles etc. will be drawn.
- **Axis Colour.**
The colour of the X-axis and its labels.
- **Info Window Colour.**
At the bottom of the Trend object is a window in which the signals being trended are listed along with their current values. In historical Trend mode this window displays the minimum, mean and maximum values of each signal at the current cursor position. The designer uses this property dialog to specify the colour of this window.
- **Font.**
The font used on the trend may be selected by pressing the **Font** button.
- **Time Span.**
This is the default time span that the Trend object covers. The time interval should be specified in the standard Prodigy timespan format, which is Xh Ym Zs, where X, Y and Z are numbers and h, m and s represent hours, minutes and seconds respectively. For example 3h 10m 25s means 3 hours, 10 minutes and 25 seconds. When running, the operator can adjust the time span.
- **Number Of X-Axis Ticks.**
Specifies how many ticks appear on the X-axis. There is always a tick at the left of the X-axis (co-

incident with the Y-axis crossing) and the number of X-axis ticks specifies a number in addition to this. In other words, there will always appear to be one more tick than requested.

- **Use Interval.**

If you click this box then instead of specifying the number of tick marks to display you specify the tick mark interval. For instance if the Time Span is 3m (three minutes) and a value of 30 is chosen there will be a tick mark every 30 seconds (so 7 ticks in total).

- **Show Mins.**

If this box is checked the x-axis is labelled with time strings that show hours and minutes. If it is not checked only hours will be shown.

- **Show Secs.**

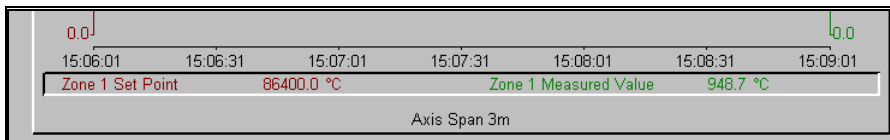
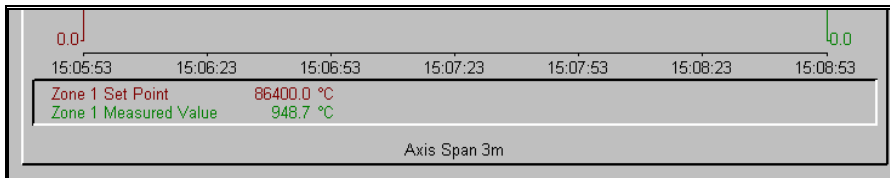
If this box is checked the x-axis is labelled with time strings that show hours, minutes and seconds. If it is not checked only hours and minutes will be shown.

- **Only Show Tagname.**

Checking this option will cause only the tagname to be shown on the trend. The tag description and units will not be shown.

- **Label Columns.**

The Info Window mentioned earlier contains information about the signals being trended. This property determines how many columns this information is displayed in. How many columns you choose will in turn depend on how much information you try to display in each column.



- **Allow Historical Trending.**

If this option is checked then when the Display is running the operator can double click on a Trend object to enter **Historical Trending**. The Trend object expands to full screen size and the data for each signal is read from the history file. When in Historical Trend mode the operator can scroll backward in time to look at data from any time period and can also select batches, perform smoothing, export data, etc. This is explained in more detail later. To return to real-time trending the operator should double click on the historical Trend.

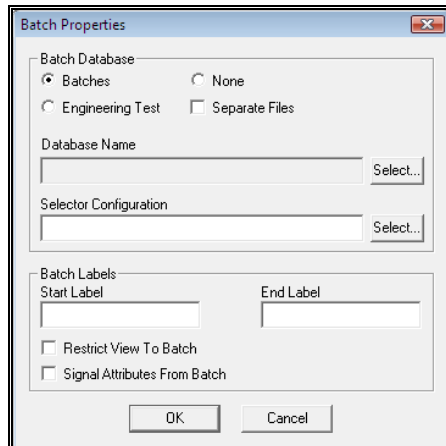
- **Only show average values.**

When in historical mode it is possible that there can be more recorded data points in a given time period than there are pixels across the screen. This means that for any given pixel position there may be more than one recorded value. In order to show this Prodigy can display the minimum, maximum and mean values that occur at a given point (if more than one value does exist). This takes up more room on the display, and is not always required. Checking this box will mean that only the average values will ever be shown, saving space on the display.

- **Permanent Historical Trend.**

A Trend can be made to be a permanent historical Trend. Its operation is exactly as described above except that it always occupies full screen. Because of this, only one permanent historical Trend should be on a given Display Builder display, and no other Display Builder object should be used on the same display.

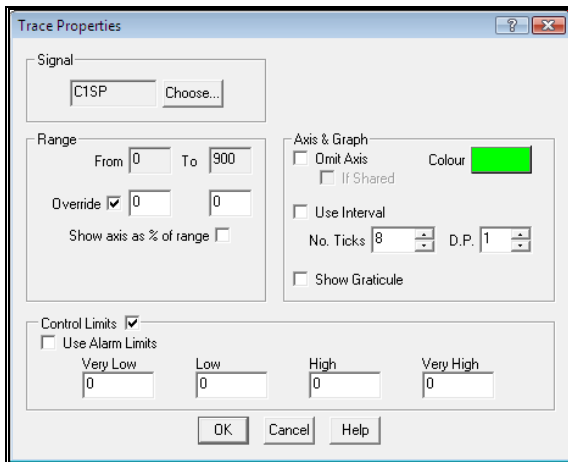
- **Batch Options.**
Press this button to show the Batch Options dialog (below).



- **Batches/Engineering Test/None.**
Select the relevant database type. Engineering Test is a specialist option and is not generally available. Select *None* if this trend will not be allowed to access batches.
- **Separate Files.**
When a batch is created, it is added to the batch database specified in the relevant field below. Alternatively you can specify that each batch is to be created in a separate database file. This option is only really viable when you are not expecting to make reports that perform analyses across many different batches (e.g. weekly or monthly totals, etc).
- **Database Name.**
If you have not checked the *Separate Files* box, pressing the *Select* button will display a dialog that allows you to select the database file that the batches will be written into. If you **have** checked the *Separate Files* option, pressing the *Select* button will allow you to choose a directory into which all batch files will be written (the batch files will have the same name as the batch).
- **Selector Configuration.**
Specify a *Batch Chooser Configuration* file (see elsewhere in the manual for more about Batch Chooser Configuration).
- **Batch Labels.**
Enter the labels to be used to mark the start and end of a batch on a historical trend. By default these are “Batch Start” and “Batch End”, but you may want to use something different, for instance “Study Start” or “Test Start”, etc.
- **Restrict View To Batch.**
Checking this box will have the effect that the user cannot step *entirely* outside of the currently selected batch when stepping through the data. Either the **Start Batch** or **End Batch** marker will always be visible.
- **Signal Attributes From Batch.**
If this box is checked and signal attribute data has been recorded by **BDR** (see the relevant manual section), the attributes such as signal description, signal range and alarm levels are obtained from the batch and used rather than those currently in place for the signal. This means that different batches can have different signal attributes, reflecting those that were in use at the time the batch was recorded.
- **Trend/Barchart**
A real-time Trend can be displayed as a stacked barchart, each bar displaying the current value for the signal it represents. This is particularly useful for showing profiles.
- **Runtime Configuration Saving.**
At runtime it is possible for the operator to make temporary changes to a trend – adding or removing signals, changing colours, overriding scales and so on. Whether or not these changes can be saved to the permanent copy of the Display Builder configuration is decided here. Check the **Allow** box if it is to be allowed, and optionally specify a privilege that must be obtained.

Trace Properties

Each signal that is included in a Trend has a set of properties associated with it. Select the relevant signal from the scrolling list in the Trend Properties dialog and press the “**Properties...**” button to display the **Trace Properties** dialog (or double click the item in the list).



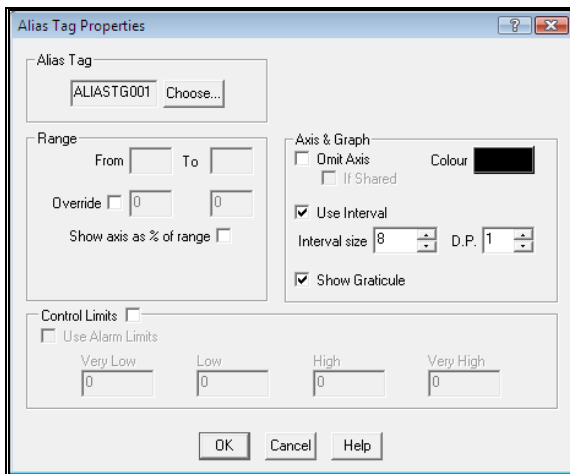
The properties that this dialog allows us to set are:

- **Signal.**
The current signal name is shown and a button provides access to the Signal Selection dialog to allow a different signal to be chosen.
- **Range/ Override.**
The range governs the extent of the Y-axis. By default this is taken from the signal range set up in the signal database, but the designer is at liberty to override this value by checking the **Override** box and entering values into the two fields to the right of this.
- **Show axis as % of range.**
Checking this box means that the Y-axis will be shown as running from 0 to 100%. The signal value is calculated as a percentage based on the signal range being used, either the default or the override, if selected.
- **Omit axis/If Shared.**
When many graphs are being displayed that share a common range, it can be useful to show them all against one axis, thus making more space available for the actual graphs. Checking the Omit Axis box causes no axis to be drawn for this signal. If the **If Shared** box is checked the axis will only be omitted if another axis that is not currently omitted can be found with the same range. This is especially useful with the axis-dragging facilities that trending offers.
- **Axis colour.**
Sets the colour not only for the Y-axis of this graph but also for the graph itself and the information about this signal that is displayed in the information window at the bottom of the Trend display and the control limit lines.
- **No. Ticks.**
Sets the number of tick marks that will be shown on the Y-axis. This number does not include the tick mark at the lowest Y position, co-incident with the X-axis crossing.
- **Use Interval.**
If this box is checked then instead of specifying the number of tick marks you specify the tick interval instead. If the axis runs from 0 to 850 and you specify a tick interval of 100 there will be ticks at 0, 100, 200 up to 800. The final axis value of 850 will be unlabelled in this instance, but would be if it coincided with a tick increment.
- **D.P.**
This controls the number of decimal places that will be displayed for values of this signal, both on the Y-axis labels and in the information window. The range for the number of decimal places is 0 to 4.
- **Control Limits.**
The control limits are four dashed lines that are drawn horizontally across the graph at the

Y-positions specified and in the same colour as this particular signal. They provide a visual aid in finding where signals fall outside of defined limits. You may choose these limits manually or you may check the **Use Alarm Limits** box to have the system automatically provide values based on the alarm limits specified in the signal database. If this latter option is chosen, the values in the fields will be those of the alarm limits. Where no alarm limit is specified in the database, a value of zero will be placed in the relevant box on this dialog, but no line will be drawn on the trend, not even at zero.

Alias Tag Properties

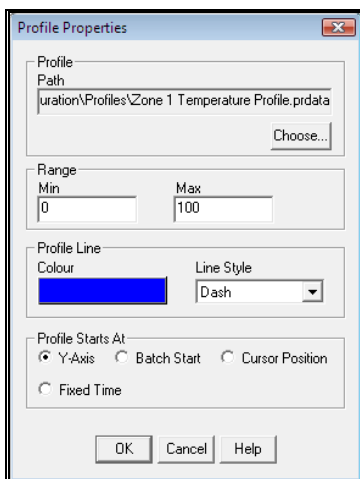
Each alias tag that is included in a Trend has a set of properties associated with it. Select the relevant tag from the scrolling list in the Trend Properties dialog and press the **“Properties...”** button to display the **Trend Properties** dialog (or double click the item in the list).



The properties are exactly as for signals. Where the range or limits are to be taken from the signal database, this is done at run-time when the alias tag is de-referenced.

Profile Curve Properties

Each profile that is included in a Trend has a set of properties associated with it. Select the relevant profile from the scrolling list in the Trend Properties dialog and press the **“Properties...”** button to display the **Profile Properties** dialog.



The properties that can be set are:

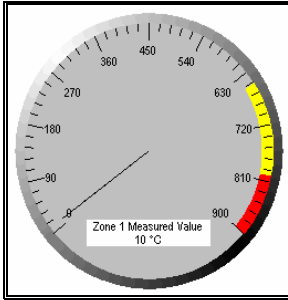
- **Path.**
The path of the profile definition. Press the Choose... button to change the path.

- **Range.**
Enter the range over which the y-axis of the signal will be drawn.
- **Line Style.**
Allows you to select the colour and line style of the profile curve.
- **Profile Starts At.**
Indicates where drawing of the profile curve should begin. Profile curves are defined as a series of points at discrete times relative to 0. The Profile Start At section of this dialog indicates where time 0 will be.
 - **Y-axis.**
Time 0 is coincident with the current Y-axis intercept, i.e. it is the X-axis start time.
 - **Batch Start.**
Time 0 is coincident with the start time of the currently selected batch (if one has been selected).
 - **Cursor Position.**
Not available at design time. When running a Trend in historical mode, selecting this option will set Time 0 to be the current cursor position. After choosing this option and exiting the dialog the Time 0 will be set to a fixed time equivalent to the current cursor position.
 - **Fixed Time.**
This is used to show that the current Time 0 position is at a fixed time. This time will have been fixed by previously selecting Cursor Position as a start option.

Dials

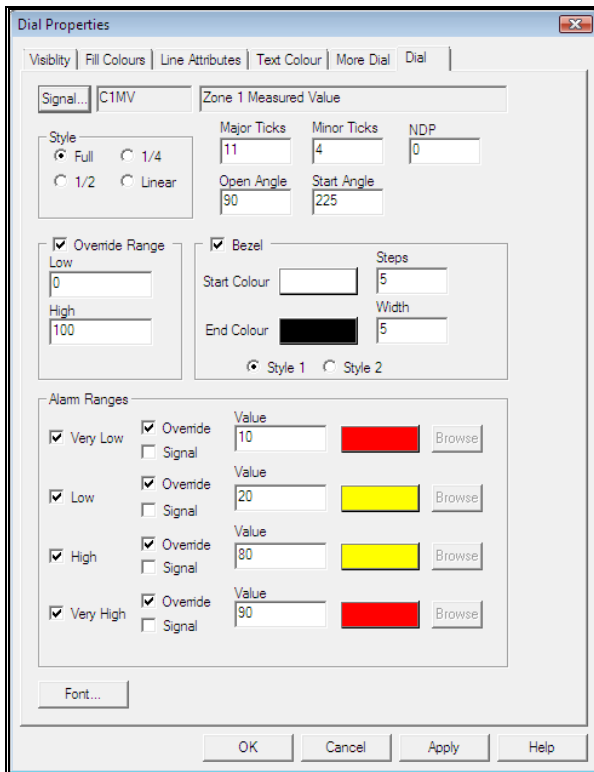
Dials are a sophisticated addition to the range of drawing tools offered by Prodigy. They give a modern and stylish appearance whilst being easy to configure and flexible in their actions.

The dial represents an analogue gauge like a speedometer or a VU meter, with a graduated scale and up to four alarm bars. The alarm bars are used to show when the signal value is getting into an alarm condition, like the “red line” area on a speedometer.



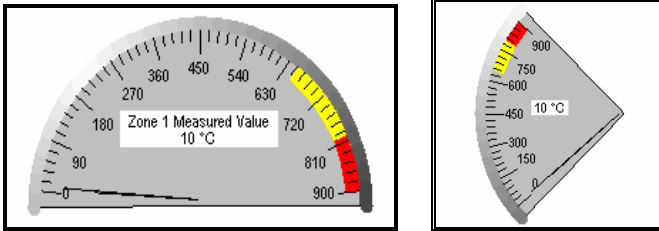
Many Dial Properties are controlled through two specialist tabs on the Properties dialog, though many standard Prodigy properties also apply to the Dial object.

Dial Properties

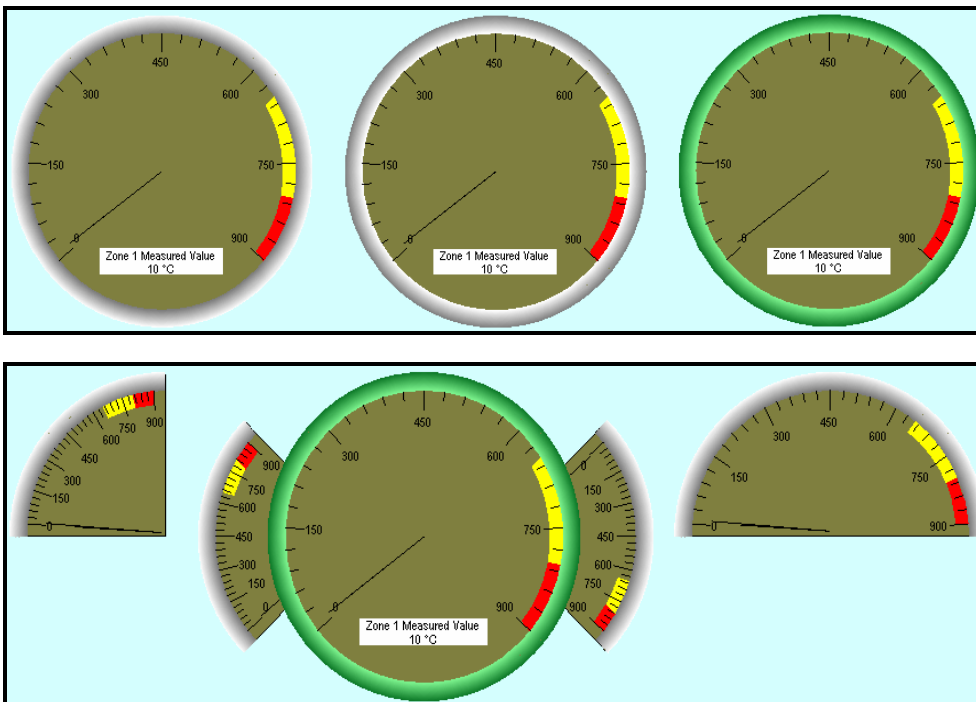


- **Signal.**
This is the signal whose value is to be displayed on the dial (by means of a moving pointer). Only real or integer signals are allowed.

- **Style.**
This is the style of dial that will be displayed. **Full** is as shown earlier and $\frac{1}{2}$ and $\frac{1}{4}$ are as shown below. **Linear** is a Bar Graph as shown later in this section.

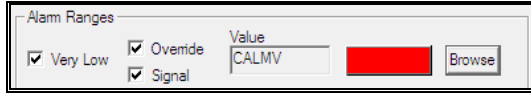


- **Major Ticks, Minor Ticks, NDP.**
These define the scale around the edge of the dial.
- **Open Angle, Start Angle.**
The open angle is the angle over which the scale is **not** displayed and the start angle is the position at which the scale begins (0 is at the 12 o'clock position). For a $\frac{1}{4}$ dial, the Open Angle is fixed at 270 degrees and for a $\frac{1}{2}$ Dial it's 180 degrees. In the example above the start angle for the $\frac{1}{4}$ dial is 225 degrees.
- **Override Range.**
The dial will take its range from the signal that is specified, but you can override this by entering custom values here.
- **Bezel.**
You can create a host of different bezel appearances by changing these parameters. Note though that the more **Steps** that you choose the slower the draw time will be for the gauge.



- **Alarm Ranges.**

Alarm ranges are bars on the dial that show the very low, low, high and very high alarm bands. On the examples above only the high and very high bands are shown and are the yellow and red indicators on the dial scales. The alarm range can be taken from the alarm strategy of the signal being monitored or, by checking the override box, you can specify your own limit. In addition, you can either give a fixed value for the alarm level or have it taken from a signal, as shown below.

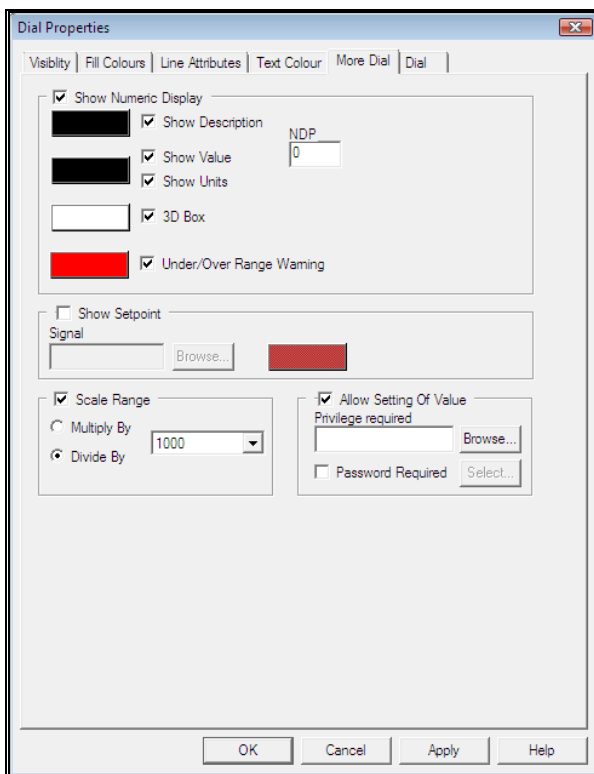


- **Font.**

Click this button to select the font used for all text on the dial.

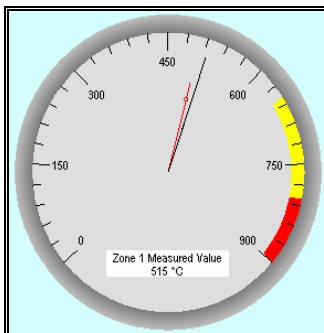
More Dial Properties

There is a second page of Dial Properties.

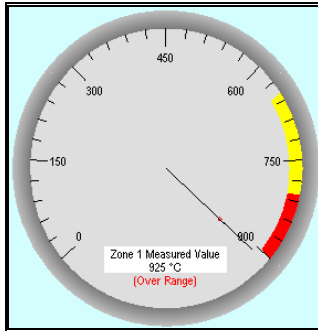


- **Show Numeric Display.**

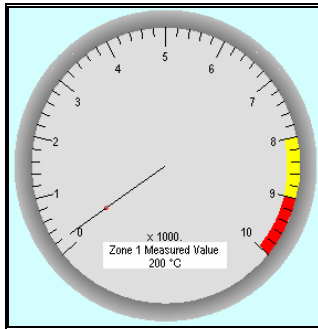
The Numeric Display is the little box inside the dial that gives a digital readout of the value to complement the analogue value of the dial. It can also (or instead) show the description and units of the signal being logged. See the example below.



It can also show an indication when the signal value goes under or over the range shown by the dial.



- Show Setpoint.**
 Checking this box allows you to specify a second signal that will be used as the setpoint for the dial. A second pointer on the dial, having a circle towards the end of it so that it is still visible behind the value pointer, indicates the value of the setpoint. The colour of the setpoint can also be set here. In the examples above the red pointer shows the setpoint.
- Scale Range.**
 The range can be scaled if necessary, for instance if the numbers of the range are very low or very small. If the range is say 0 to 10000 you may prefer to scale the range so that it reads 0 to 10 and the dial is annotated with “x1000”.



- Allow Setting of Value.**
 Checking this box allows you to set the setpoint signal value by clicking on the dial (in run mode). The usual Display Builder Analogue Value Dialog will be displayed. If you have not checked the box for **Show Setpoint**, then instead of setting the setpoint by clicking the dial, you will be able to set the value of the signal that the dial is associated with. In either case you may specify security by giving a required privilege or password.

Other Tabs On The Dial Property Page

The **Text Colour** tab allows you to specify (and dynamically change) the colour of the text used on the scale of the dial (this same colour is used for the scale too).

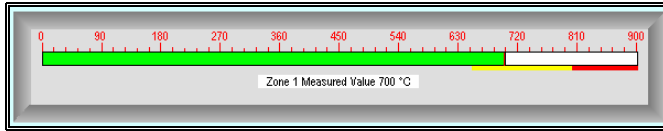
The **Line Attributes** tab allows you to specify (and dynamically change) the colour of the pointer on the dial (but not the setpoint pointer). You can also change the thickness of both the pointer and the setpoint pointer (they both use the same value) with this tab.

The **Fill Colour** tab allows you to specify (and dynamically change) the colour of the dial (not including the bezel).

The **Visibility** tab allows you to dynamically hide or display the dial depending on the value of a signal.

Bar Graphs

Bar Graph properties are exactly the same as for Dials - notice that one of the styles available for a Dial is **Linear**, which is really just a Bar Graph.

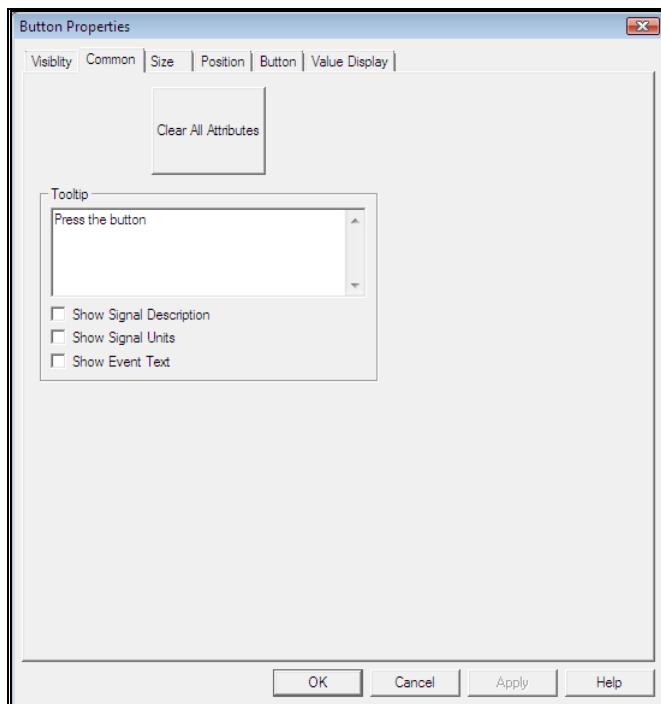


Frames

Frames give the same styles as the borders of Bar Graphs and take the same parameters in terms of the colours, width and steps used. The other tabs available with a frame allow you to dynamically change the fill colour of the frame box and to dynamically hide or show the frame.



ToolTips



Most Display Builder items can have a ToolTip assigned to them. A ToolTip is a pop-up description that becomes visible when the mouse pointer is moved over an object.



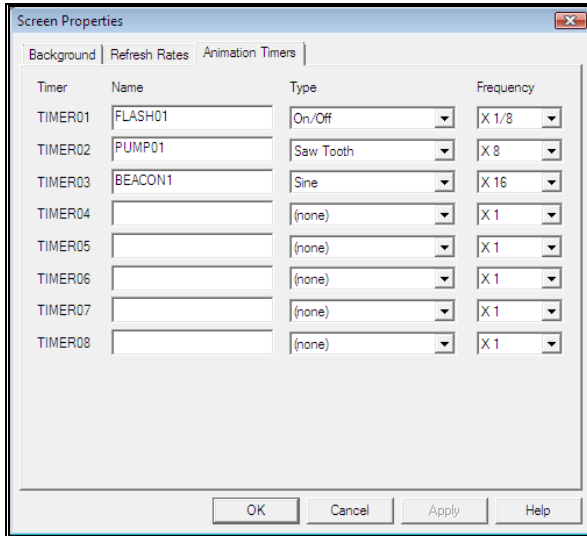
After a few seconds the ToolTip disappears.

Use it to add helpful prompts to Display Builder objects, but be careful not to over-use it in case it becomes an annoyance and clutters up the display. The Signal Description, Units or Event Text can also be included

Animation Timers

Animation Timers allow you to create dynamic objects that do not rely on signals to govern their actions. This eliminates the need to have signals tied to the evaluate driver or SLANG code whose sole purpose is to provide inputs to cause animation effects. It also makes the animation smoother because the Animation Timers are more deterministic.

In order to set up a timer, make sure no object is selected and right click on the Display background. On the dialog that appears select the Timer Properties tab.

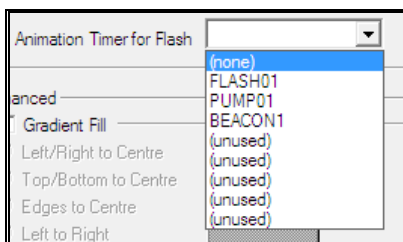


There are 4 types of timer available.

- On/Off.**
 This timer switched between an on state and an off state, and therefore only ever has the values 0 or 1.
- Saw Tooth.**
 This timer increments from 0 up to 100 and then when it reaches 100 resets to 0 before incrementing back to 100, and so on.
- Triangle.**
 This timer increments from 0 to 100 and when it reaches 100 it decrements back to 0, and so on.
- Sine.**
 This timer follows a sine wave pattern and increments from 0 to 360 (degrees), with the generated values scaled to be in the range 0 to 100.

The timers are incremented by 1 at each refresh period (see **Refresh Rates** tab) but you can scale the increment of the timers by adjusting the **Frequency** count. If you increase the frequency (X2, X4, etc) the counter will be incremented by 2, 4, etc instead of by 1, resulting in a faster movement. If the frequency is decreased (X 1/2, X 1/4, etc) then the counter is incremented by 1 every 2nd, 4th, etc cycle of the refresh timer. Note that the On/Off timer can only be slowed down – it obviously cannot flash more rapidly than the refresh rate of the Display.

You can specify up to 8 different timers and give them names. When you select timers they appear in the list using either the name that you give them or (if you leave the name blank) as a default name based on the type of timer. To stop using a timer, simply select the “(none)” option from the list.



Making A Better Display

Dynamic Tag Substitution

Sometimes a system will have a set of Display Builder displays that must be repeated several times because there are many process lines that are the same. Essentially the only difference between the Displays is the tags that are attached to the objects in the display. It would be wasteful and time consuming to have to create and maintain all of these Displays, so Prodigy provides a facility called Dynamic Tag Replacement. This allows you to create just one set of Displays (including trends) and then have this Display dynamically swap the tags that are used in them.

The way that this works is that a substitution string is provided to the Display when it is invoked. This string consists of the first n characters of the tag name, and each tag used in the Display has its first n characters replaced by this string. You can see that in order to use the technique successfully, your signal database must be created with this in mind. For example, if we have two furnaces and each has a setpoint and measured value, we would create a database with signals called F1SP, F1MV, F2SP and F2MV. We would create one Display that used the tags F1SP and F1MV (though any combination would be valid). We then invoke the Display with the substitution string F1 or F2, depending on which furnace we want to see.

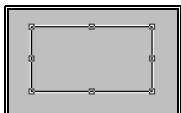
To invoke Dynamic Tag Substitution from the Menu Bar, simply add the parameter `-replace="abc"` to the parameters being used to launch the Displays (*abc* would be the substitution string **F1** or **F2** in our example above). To invoke it from another Display, simply check the relevant box on the **Button Actions** dialog and enter the substitution string. See the **Button Actions** section of the manual, above.

Once a substitution has been made it stays in effect for any other Displays that are launched from buttons on that Display. To cancel a substitution (for instance to return to some main overview Display that has no substituted tags) check the **Substitute** box and leave the substitute string blank.

Making Composite Objects

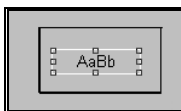
Placing these individual objects on the screen and assigning them attributes makes for a pretty simple looking Display. In order to make the Displays more visually appealing you need to construct a Display object from several basic objects. For instance, adding highlight and shadow lines to create a bas-relief effect. Suppose we want a faceplate that is rectangular with a numeric value displayed upon it.

First create a rectangle object:



Then add two line objects, each consisting of two segments. The first runs bottom left to top left to top right, the second from top right to bottom right to bottom left.

Then add the text object:



- In the rectangle object properties set the fill colour (here it is dark grey).
- In the text objects properties set the signal whose value you want to monitor, the formatting code and the required font. Set the text colour too (here it is yellow)
- In the line object properties set the bottom left to top right line segments colour to be white and its thickness to be 3.

- Set the other line segments colour to be black and its thickness 3. This gives us the bas-relief effect.



This is how it looks when we run.



When selecting objects that are co-incident or close to each other a special technique must be used. Left clicking at a point while holding down the control key causes the next object at the same location to be selected. In our example clicking in the centre of the area occupied by the object whilst holding down the control key will select the bottom right line, then the top left line, then the text object and then finally the rectangle. Right click over the background, not over an object, to bring up the property sheet for the selected object.

Grouping Objects

Now that we have created a bas-relief display panel, we can group the objects that form it together so that it can be moved, duplicated, stored or even deleted as if it were one object. In common with most other drawing packages multiple objects can be selected by selecting the first one and then clicking on the others in turn while holding down the shift key. Alternately a marquee box can be dragged around the items. Any items intersecting the marquee will be included in the group selection. If required you can even select all of the objects in the Display by choosing **Select All** from the edit menu.

Having selected all the objects that form a group, simply click on the **Group** button (below) on the formatting toolbar at the bottom of the screen or press control-G.



The object may now be moved around as if it were one. To change the properties of one of the components of a grouped object, use the control-click technique described earlier or ungroup the objects.

Ungrouping Objects

Select the grouped object and press the **Ungroup** button (below) on the formatting toolbar or press control-U.



Grouped Attribute Changes

You can change the (similar) attributes of all objects in a group in one go. Similarly, you can select multiple objects (by shift-clicking) and change the attributes of all of those objects in one go.

Select the object(s) in question and then right click on one of them. The usual **Properties** dialog will be displayed. Make changes as usual but note the following points, and note them well.

- The fields shown in the tabs on the dialogs are simply those from the **last** object that was selected, they in no way imply that these values are currently used for all of the selected objects.
- If **no** changes are made to anything on a tab, **none** of the attributes on that tab are applied to any object. So, you can make changes to things on the Orientation tab, and attributes on the Size tab etc will not be applied.
- If you make a change to **any** attribute on a particular tab then **all** attributes from that tab will be applied to **all** of the selected objects. So, if you alter just the **Very High** value on a **Fill Colour** tab, all of the other values on that tab will be applied too.

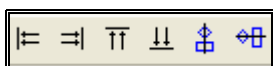
- On the **Position** and **Size** dialogs, horizontal and vertical movement or scaling are treated separately. So you can change a horizontal parameter without affecting vertical and so on.
- The **Signal** attribute and the check box that determines if the feature is being used are **never** applied to **any** of the objects in the group.
- All common attribute tabs are shown, so **Value Display** and **Text Colour** will be available even if no Text objects are selected. Making changes to these tabs in those circumstances will have no adverse effect (but will obviously be a waste of time).
- The attributes for **Dials/Bar Graphs** and **Frames** are **not** currently displayed as part of a group attribute change.
- If you make grouped changes and then **Undo** them, the undo is performed object by object by object and not to all objects at once. You must repeatedly press **Undo** until all objects are undone.

Duplicating Objects

Select the object to be duplicated and then click the **Duplicate** button (below) on the formatting bar or press control-D. The new object will overlay the existing object, offset by a few pixels. Drag it to its required position. All attributes, including the signals selected, are copied from the old object.



Aligning Objects



Often several objects may need to be aligned in order to improve the neatness of the Display or to ensure coincidence of positioning. Six buttons on the formatting bar provide alignment functions. Select all of the objects to be aligned, being sure that the last one selected is the one you want to align the others to, this being known as the dominant control (its handles are a different colour than the other objects, to highlight its dominance). Press the relevant alignment button, these being, from left to right

- Line up left edges
- Line up right edges
- Line up top edges
- Line up bottom edges
- Centre horizontally (i.e. horizontal position is affected).
- Centre vertically (i.e. vertical position is affected).

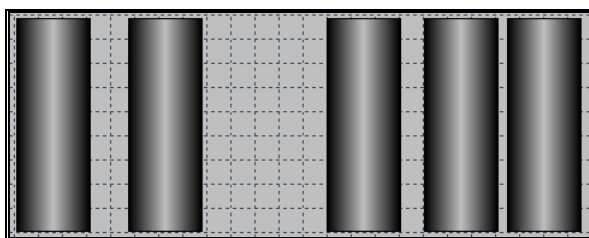
Spacing Objects



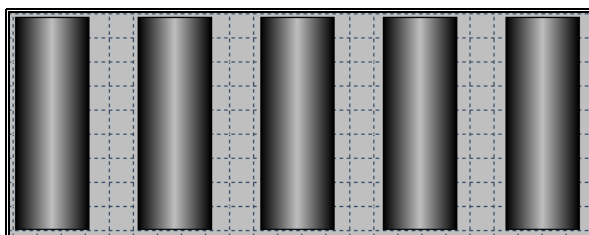
At times you may wish to space several objects evenly across a display. To do this, select all of the objects that you wish to space and use one of the two buttons on the formatting bar that are show above. The left button spaces objects across the screen, the right one spaces them down the screen.

To space objects across the screen, Display Builder takes the left and right extremities of a bounding box that would contain all of the objects. Any objects that are at these extremities remain unaffected and the other objects are positioned so that there is an equal sized gap between all of the objects being spaced.

Before...



And after.



Spacing objects vertically works in exactly the same way.

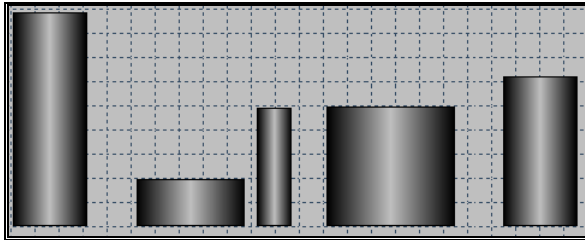
Sizing Objects



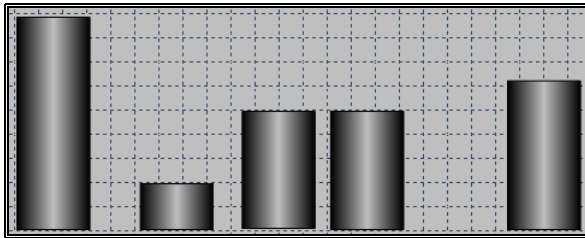
Sometimes you may want several objects to be the same size as each other. You can do this by using the three formatting buttons shown above. Select the objects to be sized, ensuring that the last one selected – the dominant control – is the one you want all the other objects to be the same size as. Then press the required button. The function of the buttons is, from left to right:

- Make objects same size
- Make objects same width
- Make objects same height

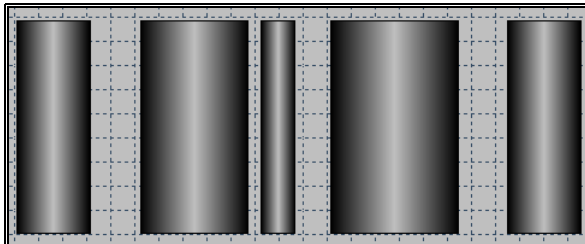
All different...



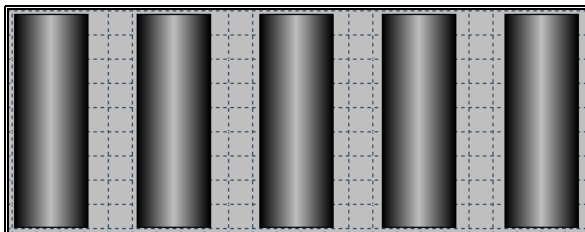
Same width...



Same height...

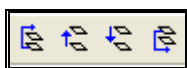


Same size.

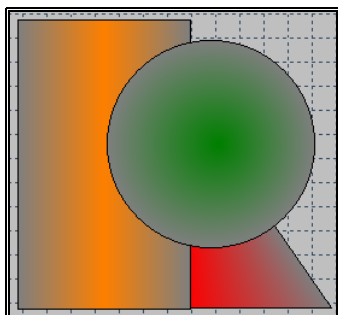


Use the alignment or spacing functions to line up the objects as required.

Changing the Z-Order

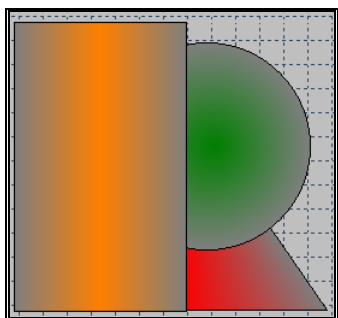


When building up composite objects or Display Builder displays some objects must appear *behind* or *in front* of other objects. Often several objects will appear layered over each other. Where an object appears in this stack is known as its *Z-order*.

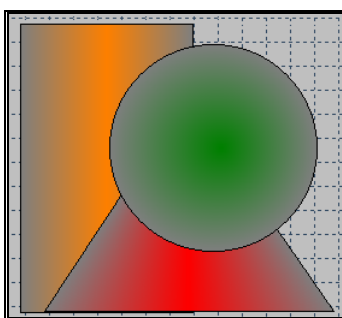


In this picture the circle is at the top of the Z order, followed by the rectangle and finally the triangle. To alter the Z-order, use the buttons shown above. From left to right these:

- Move an object to the top of the z-order
- Move an object up one in the z-order
- Move an object down one in the z-order
- Move an object to the bottom of the z-order.



Circle down one in z-order

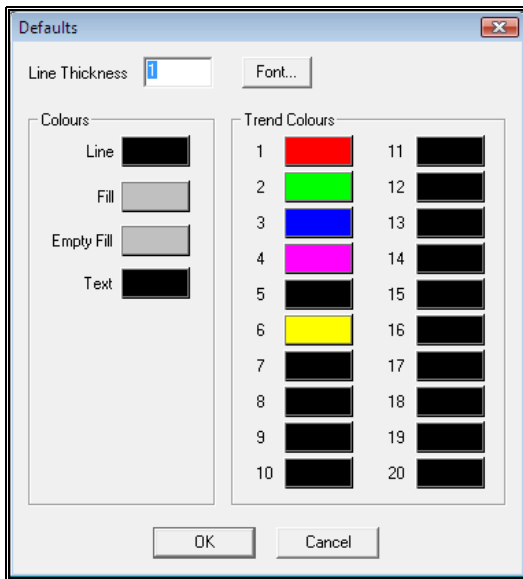


Rectangle to bottom

Setting Defaults

You can set default values for **Line colour**, **Fill colour**, **Empty fill colour**, **Text colour** and **Trend Colours**. In addition, you can specify a default **Line thickness** and **Font**. These values will then be automatically inserted into each new object that is created.

To change the default values for objects you click on the main file menu and select system defaults, this will bring up the dialog below and allow you to change all the default properties assigned to objects.

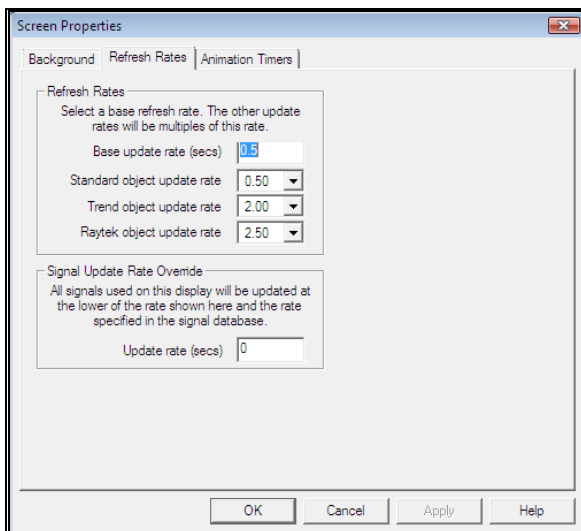


- Click on the relevant colour buttons to set the various default colours required.
- Click on the **Font** button to set the default font.
- Enter a value for the default line thickness.
- Click OK

Altering Refresh Rates

All objects in Display Builder are refreshed at a given rate. By default, *standard* Display Builder objects are refreshed every ½ second, *Trends* are refreshed every 2 seconds and *Raytek* heat map objects are refreshed every 2½ seconds. Display Builder allows this refresh rate to be adjusted.

Make sure that *no* Display Builder objects are selected (click on the background) and then right-click. The *Screen Properties* dialog will be displayed.



- Select the **Refresh Rates** tab.
- Set the **Base update rate** – all of the other update rates will be multiples of this. The base update rate must be between 0.05 and 3 seconds.
- Set the update rates of the **Standard**, **Trend** and **Raytek** objects. Select these from the drop-down lists.

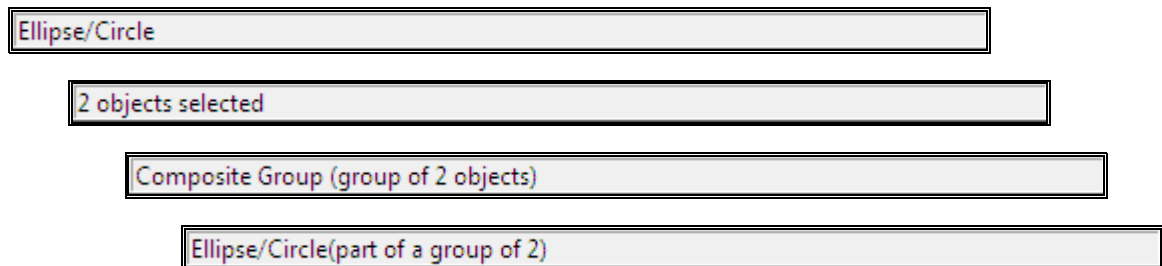
- If desired, set the **Signal Update Rate Override** period. This is a “safety net” to make sure that all Display Builder objects are periodically refreshed with new values. It should not generally be used, it is just for those circumstances where periodic queue filling can occur.

Beware:

Changing the refresh rates of the objects can give much smoother looking animation effects. However, setting the refresh rate to too small a value can cause Display Builder to become processor intensive. In effect, it spends all its time constantly redrawing objects. Be particularly careful with Trend and Raytek objects and always check the amount of CPU time Display Builder is taking before committing to a final value for the refresh rates.

The Status Bar

The status bar at the bottom right of the Display Builder display provides information as to what and how many objects are currently selected:

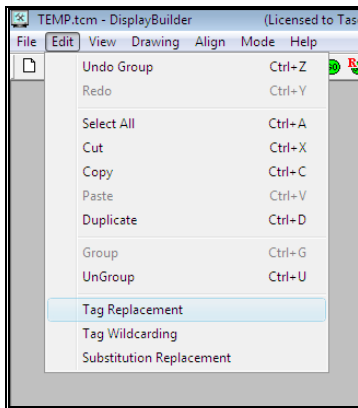


Finding and Replacing Tags Used in a Display

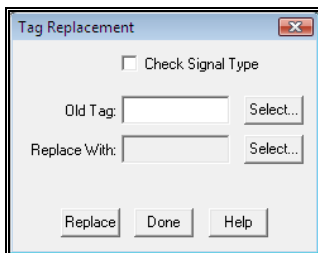
Suppose that you take a copy of a Display that is used on one process and you want to use it as the basis as a Display for another process. If you are lucky, it may in fact be identical in form other than requiring changes the tag names that are used. Prodigy provides a way of changing the tagnames that are used in a Display, either on an object by object basis or globally.

Changing the Tags Used in an Object

There are two ways to change the tags used in a Display Builder object. The first is simple Tag replacement - select the object for which you wish to do a tag replacement by left clicking on it. From the Edit menu select Tag Replacement.



This will display the following dialog.



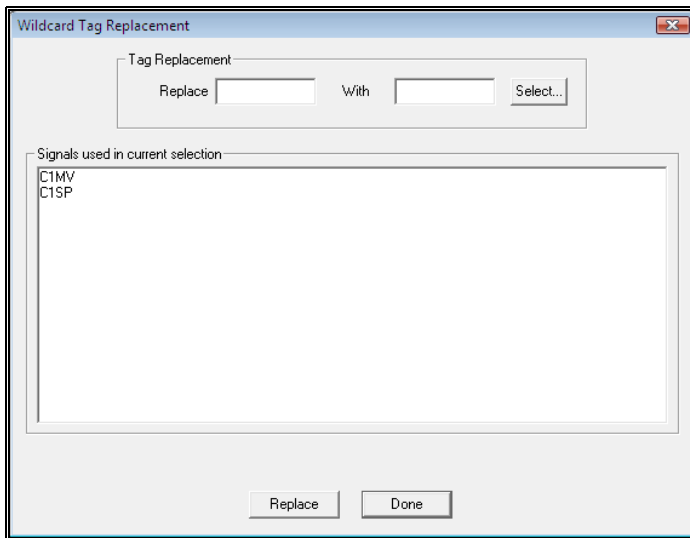
- Enter the name of the tag that you want to replace into the box labelled **Old Tag** or select the required signal by pressing the **Select...** button to the right of the box.
- Select the new tagname to be used from SigScan by pressing the **Select...** button to the right of the **Replace With** field.
- Press the **Replace** button when you are ready to perform the replacement.

A message will appear above the **Replace** and **Done** buttons that tells how many tags were replaced.

As an aid to data integrity, you can check the **Check Signal Type** box when doing tag replacements. When this option is checked both tagnames must be selected via SigScan and the system will not allow the type of the **Replace With** tag to be different from type of the **Old Tag**.

*When selecting the **Old Tag** SigScan will show tagnames that no longer exist in the signal database. This is because it is probable that you will want to alter any tags that are no longer in the database to ones that are. However, if you have checked the **Check Signal Type** box and you select a signal that is no longer in the signal database you will get an error message. Obviously the system cannot check the type of a signal that no longer exists.*

The second option is **Tag Wildcarding**. Select the object(s) that you want to substitute tags in and then select **Tag Wildcarding** from the Edit menu. The following dialog will be displayed



The dialog shows a list of signals used in the currently selected object(s). You can do wildcard replacements on these signals by using the * character to mean “any characters”. For instance, in this example if we chose to replace C1* with C2* then C1MV would become C2MV and C1SP would become C2SP.

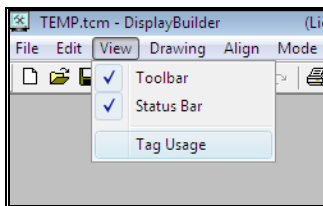
The signals C1SP and C2SP would have to exist in the database for this to occur – tag wildcarding will not create signals that do not exist. If, for example, C2MV does not exist then the tag will remain as C1MV.

Changing Tags in Multiple Objects

Select all of the objects in which you want to change tags by doing a multiple select or a **Select All**. Then follow the procedure described above. The tags will be replaced in all of the objects that have been selected.

Finding out Where Tags Are Used

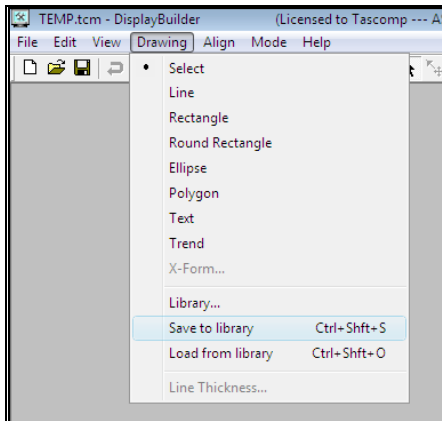
You may at times want to see where a tagname is used in a Display. To do this select **Tag Usage** from the **View** menu.



This will display SigScan from which you can choose the signal that you want to look for. On exiting SigScan, all objects that use the requested tag in any of their properties will become selected.

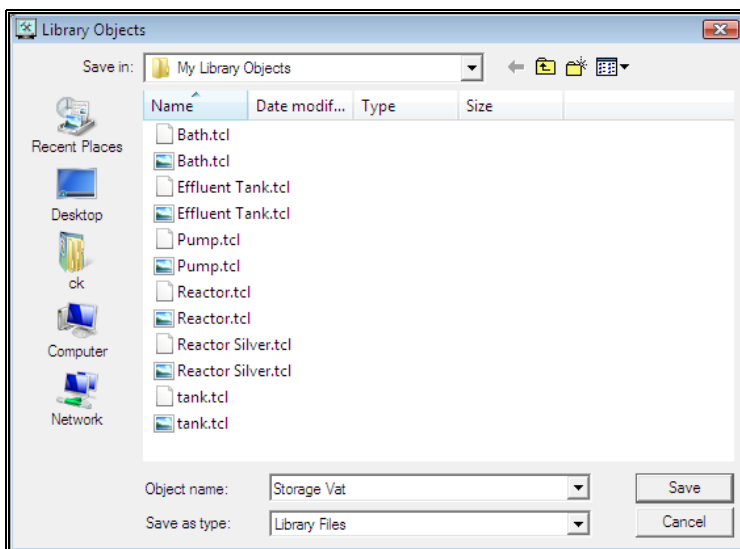
Saving Objects to the Object Library

When a good-looking Display Builder object has been created it may make sense to save it to an object library so that it can be used again in other Displays. To do this select the object (or objects) to be saved and then select “Save to Library” from the Drawing menu or press control-shift-S.



This will present a browser dialog asking where the object is to be stored (the default place is Prodigy\Library\My Library Objects) and the name under which it will be stored.

Create relevant directories, e.g. Faceplates, Vessels, Pumps, etc and save the object with a descriptive name e.g. “Storage vat”.

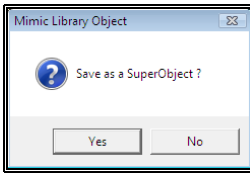


SuperObjects

When the object is saved, Prodigy also stores information about any signals that have been used within the object. This information includes the signal name, description, units, range, driver, address, etc. When this object is subsequently loaded again, it will retain all of the tag information. Prodigy will give you the option to use an existing signal in its place or to create a new signal having all of the attributes that have been stored.

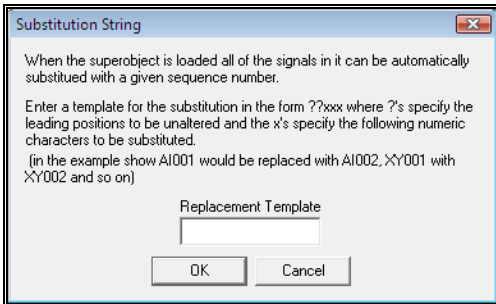
This facility is known as SuperObjects, and Prodigy contains a library of pre-configured SuperObjects that you can load and modify as required.

When you save an object to the library, and if it contains attributes that use signal values, you will be prompted as to whether or not you want to save it as a SuperObject.



If you choose No the object will be saved but no signal attribute data will be stored for that object.

If you choose Yes then the signal attribute data will be stored, and the system will also prompt you for a **Replacement Template**. This allows automatic substitution and incrementing of signals specified in the SuperObject.

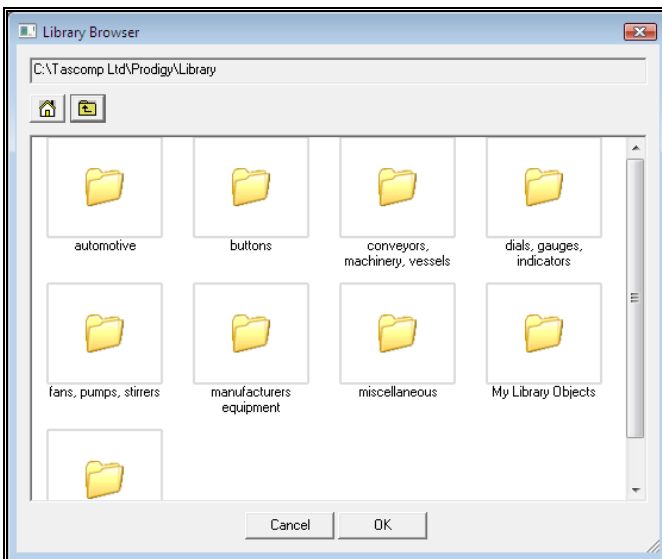


The template takes the form ???xxx, where ?s are the characters in the tagname to leave unchanged, and xs are the digit positions to be filled with an incrementing number. For example, if the template was ??xx and there was a signal AI01 in the SuperObject, then successively adding this object would create the signals AI01, AI02, AI03 and so on. If AI04 already exists, then adding a new SuperObject would create AI05 – in other words adding a SuperObject would always create a new, unique signal.

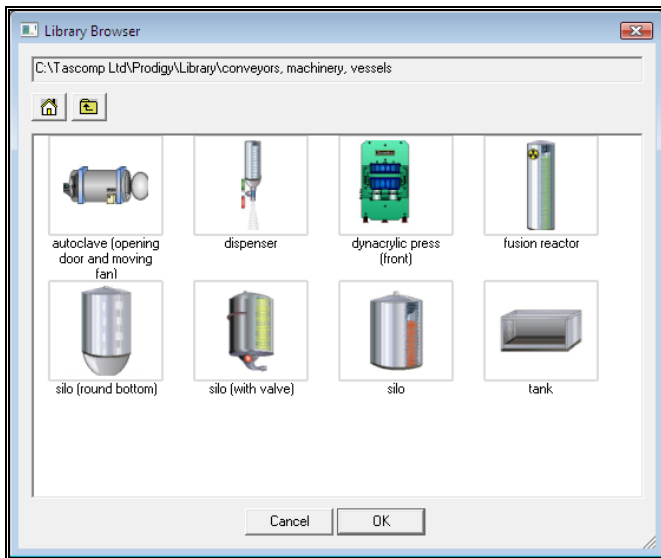
Loading Objects from the Object Library

Select the **Load from Library** option from the Drawing menu or press control-shift-O. This will present a dialog that allows you to choose the relevant object. Prodigy Library Objects are shown in the browser as a little thumbnail of the library object, to make selection easy.

The browser will open in Prodigy\Library – select the relevant directory where the library object that you want resides (in this case My Library Objects\Vessels).



Select the required object.

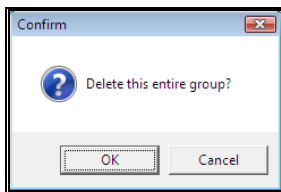


If the object is a SuperObject you will be presented with a dialog that allows you to configure the signals within the SuperObject. This is explained in detail later.

Once loaded, drag the object to its required position in the Display.

Deleting Objects

Select the object to be deleted (it may be a grouped object) and then press the delete key (not the backspace key). The object will be deleted. If it is a grouped object it will first prompt as to whether you really want to delete it.



If you choose **OK** the whole group will be deleted.

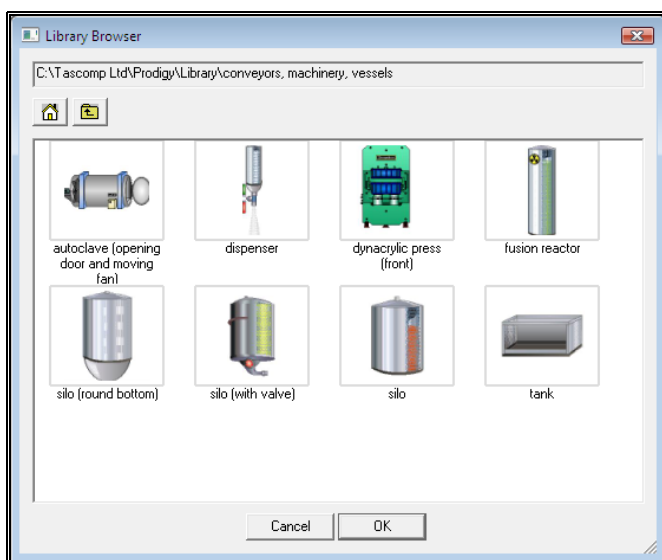
Saving a Display

This is as for any document based windows program. The File menu gives options for "Save" and "Save as". Make relevant subdirectories to store your Displays in a sensible fashion, for instance one directory per plant area.

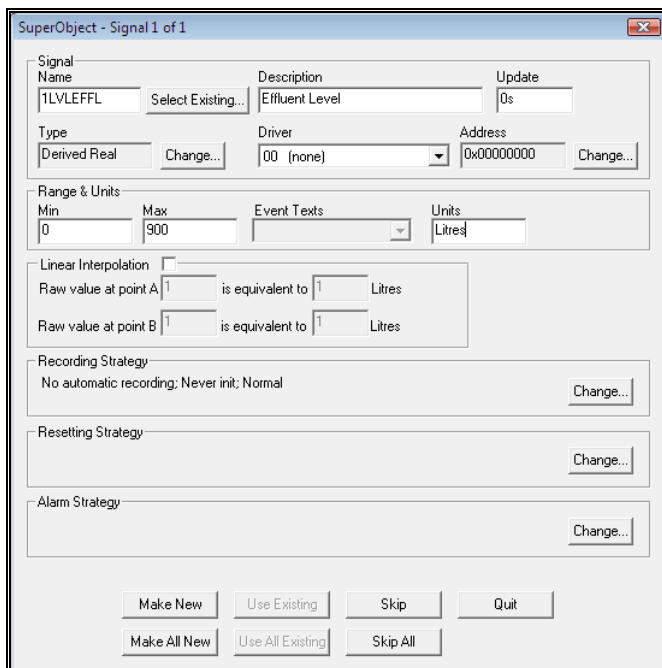
SuperObjects

When a library object is added into a Prodigy display it might be a SuperObject. A SuperObject is a library object that has signal information embedded within it. Basically, all of the tags that are required to define the object's behaviour are embedded within the object, whether these are input signals that control the object's colour, size, position, value display etc, or if they are outputs like the setpoint of a controller. In addition, attributes of the tags are also stored, such as their units, range, type and so on.

SuperObjects are highlighted in the Library Browser by having a thicker, coloured border drawn around them.



When a SuperObject is placed onto a display Prodigy can create those signals required by it from the information embedded within it. The user is presented with a dialog that gives them the option to either allow Prodigy to create a new signal or to select an existing signal to use in place of the tag in the SuperObject. This dialog is shown below.

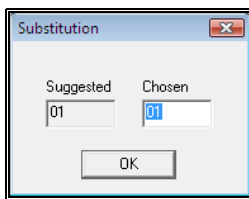


There are a lot of fields and options on this dialog.

- The simplest way of using it should just be to click the **Make New** or **Make All New** buttons, which will tell Prodigy to create a new signal in the Signal Database for each tag that there is in the object.

- Alternatively if you want to use existing tags instead, then you will need to select the correct signal that you want to use in place of the tag in the object. The *Use Existing* button is not available until a valid, existing signal is selected in the **Name** field at the top of the dialog.
- Finally you can create new signals on a one-by-one basis, changing any of the suggested attributes as you go. This is the most complicated and time consuming method of using SuperObjects, but it still offers a great saving over having to configure the database as a separate exercise from creating the object on the display.

If the SuperObject was stored with a substitution template for the signals embedded in them, suggested new signal names will be generated automatically based on this template. The system will find an incrementing number that will allow the generation of a complete set of unique signals for the template signals used in this object. For instance, the system will try “1” first and it substitutes a 1 into all of the signals in the SuperObject. If any one of signals created already exists, then it will try “2” instead, and so on until a set of unique signals is created. At this point the system will present a dialog as shown below.



This shows the incrementor value that the system suggests but you can alter this to something of your choice. If the value that you choose will not allow a set of unique signals to be created, the dialog will appear again with the next number greater than the one you entered that can be used as a valid incrementor.

Once an incrementor is selected, the large SuperObject properties dialog (shown above) is displayed so that you can alter or accept the attributes as necessary.

Using A SuperObject

Place a library object onto a Prodigy display and if it has embedded signal information the dialog shown above will be displayed. The fields and options on this are described below.

- **Signal Name.**
Enter the name that this signal is to have. The default is the name that the signal has within the SuperObject. If the name that is selected is the name of an existing signal, all fields are greyed out along with the *Make New* and *Make All New* buttons. The only options available are to *Use Existing* or *Use All Existing*.
If the signal name does not exist, the *Use Existing* and *Use All Existing* buttons are greyed out. All other fields are on to allow any configuration changes that are required for the new signal.
- **Select Existing.**
This button may be used to select any signal that already exists in the Signal Database.
- **Update.**
- **Type.**
- **Driver.**
- **Address.**
These fields act exactly as on the *Signal Database Configuration* screen (see that section of the manual for more details).
- **Range & Units.**
Enter the signal range (minimum and maximum) in the relevant boxes, along with the units that this signal is to take.
If the signal is a *Digital* or a *Finite State* the *Event Texts* box will be available. This is a drop down list containing as many elements as the signal range dictates. To enter the event text for a particular position, select it on the drop down list and then type the text into the edit box to the right of the drop down list. The text in the drop down will change at the same time as you type into the edit box. Repeat this for each item in the drop down list.
- **Linear Interpolation.**
This section is actually the same as the signal details section of the *Signal Database Configuration*.

- **Recording Strategy.**
- **Resetting Strategy.**
- **Alarm Strategy.**
These options are as for the *Signal Database Configuration* program.
- **Make New.**
Clicking this button creates a new signal based on the information presented in this dialog. The default information should be sufficient to allow a useable signal to be created, with the proviso that the driver and signal address may need to be adjusted. Once this button is clicked the signal is created and the dialog displays the net signal that is about to be made.
- **Make All New.**
Clicking this button means that SuperObject will create a new signal for every tag in the SuperObject, based on all of the default information. No user intervention will be requested *unless* the name of the signal already exists. In this case the dialog will pause only at each signal that is causing conflict and request that you give a new name for the signal.
- **Use Existing.**
Only available when the *Signal Name* is the name of an existing signal. Clicking this button makes SuperObject use the signal name given for the current tag.
- **Use All Existing.**
Only available when the *Signal Name* is the name of an existing signal. Clicking this button makes SuperObject use the default signal name for each tag, provided that the signal already exists in the Signal Database. The dialog will pause at each signal that *does not* exist in the signal database and give you the option to select a different signal or create a new one.
- **Skip.**
Do not attach a signal to this tag. Use this option if the tag being requested is not required in your system.
- **Skip All.**
Similar to Skip but skips all remaining tags.
- **Quit.**
Exits the SuperObject dialog and abandons all changes that have been made. No signals will be created in the signal database and all tags in the object will be left unattached.

Any tags that are left unattached in a SuperObject (because the *Skip*, *Skip All* or *Quit* buttons have been used) can be attached to signals later by using the tag replacement facility within Display Builder.

Activating the New Signals

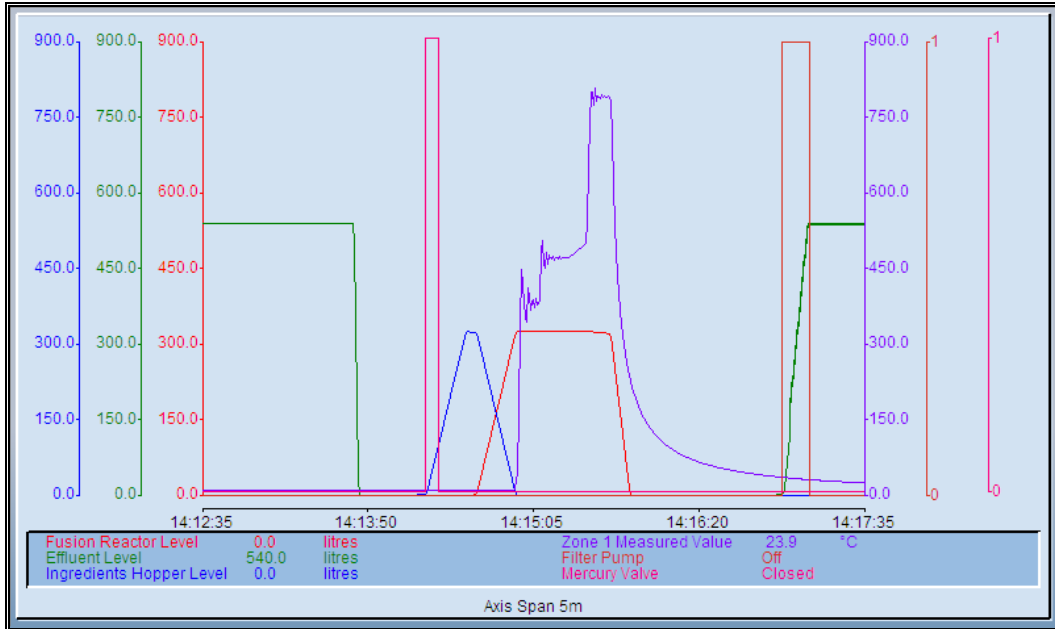
The signals *created* by SuperObject are not made available until Prodigy is restarted. Any existing signals that are used are available immediately.

Interaction with Database Configuration

If you have made changes to the Signal Database from within SuperObject, you will be warned about this fact when you access Database Configuration, and vice versa. There is nothing that you need to do about the situation really, it is just a warning so that you are aware of what is going on.

Trending

Live Trending



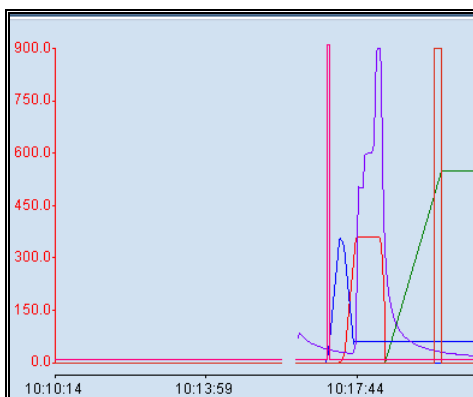
A live trend is a graphical display of signal values that updates in real time. For each second of time that passes the trace moves one second to the left and new trace values come in at the right-hand edge. If no new value for a particular signal comes in (i.e. the signal value does not change) the value is shown as a flat line, as in the green trace, above.

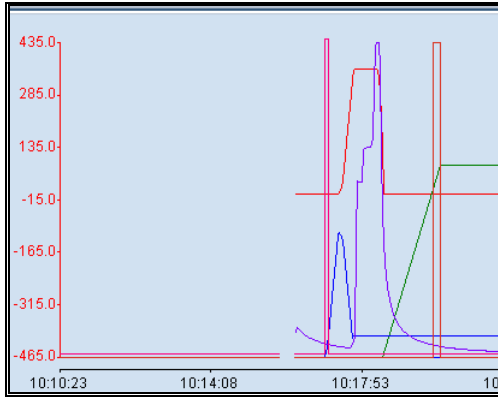
When a new value does come in the trace will be drawn from the y-point at which the last measured value occurred. In the example of the green trace this might be at the point where the line started going flat, in which case the flat section of the trace will be replaced with a line that joins the current start of the flat section with the newly occurring value. This will give a trace resembling the blue one above.

Alternatively, there may have been other actual values received that just happened to be the same as the previous value, which would also result in a straight line. In this instance the trace will retain its flat section and end give a result similar to that on the red trace above.

Axis Sliding

You can slide the y-axes while in real time trending. To do this shift-click on one of the axis tick values on the axis that you want to slide and then drag the value up or down. The whole axis will move and the trend will follow it. In the example below the red trace is dragged upwards. See how the y-axis scale goes from 0 to 900 before dragging and -465 to 435 afterwards, and the red trace is higher up the screen.



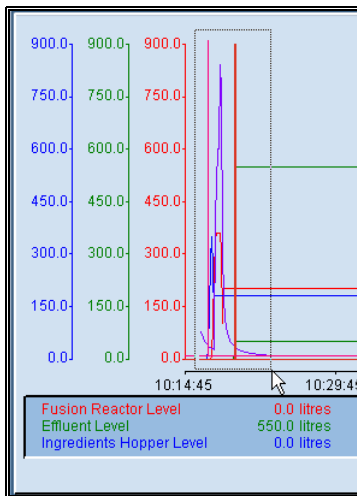


This mechanism is useful for moving overlapping traces into areas that make them easier to read.

If you have implemented Omit Axis (If Shared) and then slide an axis, it will split away from those that it was sharing with (because it is no longer shared – dragging it breaks the share). If you want to slide all axes that are in the share then use ctrl+shift+click, which will move all of the trends that share the axis being slid by the same amount.

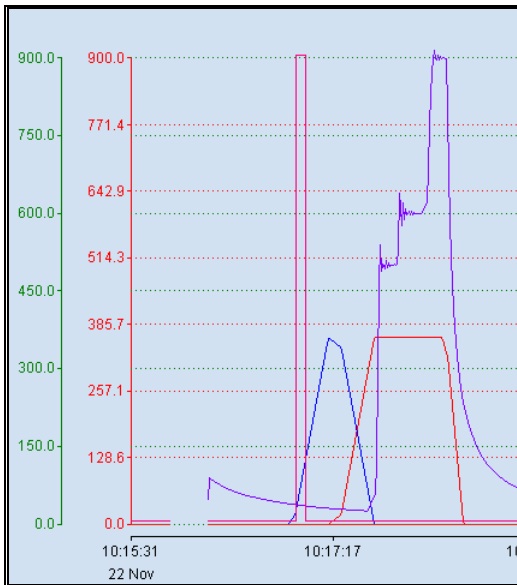
Zooming

You can zoom on a real-time trend by dragging a box around the area of interest. Zooming on a real-time trend does not affect the x-axis, only the y-axis of all traces. The example below shows a box being dragged around the traces (although the effect will be minimal in this instance as the box encloses the full height of the y-axis anyway).

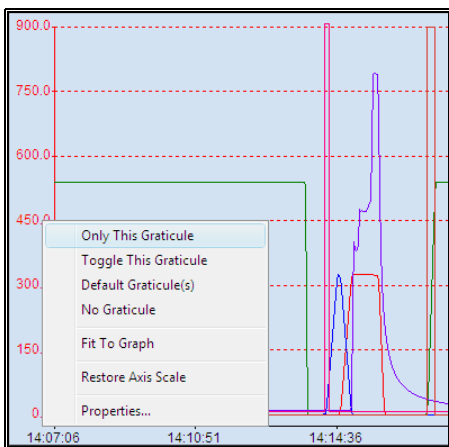


Graticules

Graticules are dashed horizontal lines that aid data read-off. Any or all of the y-axes can have a graticule displayed, though where the ticks of the axes coincide it is not really sensible to have the graticule for all such axes displayed. In the example below the graticules for the red and green traces are displayed.



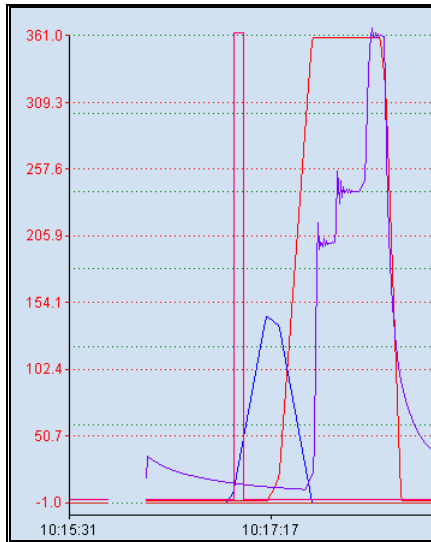
The Trace Menu



If you right-click over one of the y-axes you get a menu of options that apply to individual traces. The options on this menu are:

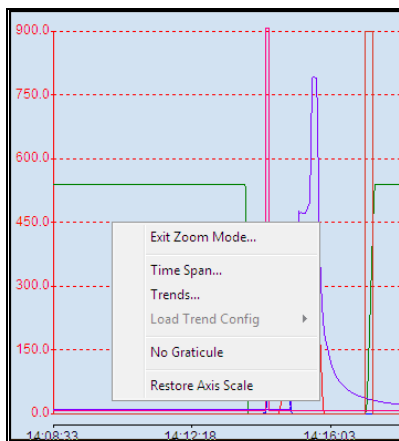
- **Only This Graticule.**
Displays the graticule for the selected axis and removes the graticules of all other axes.
- **Toggle This Graticule.**
Turns the graticule for the selected axis on or off, depending on its current state.
- **Default Graticule(s).**
Displays only those graticules that were pre-defined in the trend configuration (there may be none).
- **No Graticule(s).**
Removes any graticules that are displayed on the trend.

- **Fit To Graph.**
This option changes the upper and lower range values of the selected trace so that the trend covers the entire y-axis. The example below shows the red axis fitted to the range of the graph.



- **Restore Axis Scale.**
This option restores the range of the selected axis to its pre-configured value. This can be used to undo the effects of a **Zoom**, an **Axis Slide** or a **Fit To Graph**.
- **Properties.**
This option displays the trend properties dialog for the selected axis.

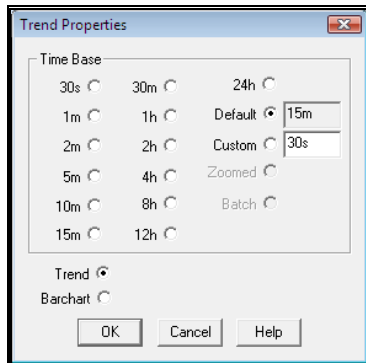
The Global Menu



This is very similar to the Trace Menu but the actions selected apply to all traces rather than just the selected one. The options are:

- **Exit Zoom Mode.**
Undoes the action of any zooming that has been done, but does not undo any **Fit To Graph** actions.

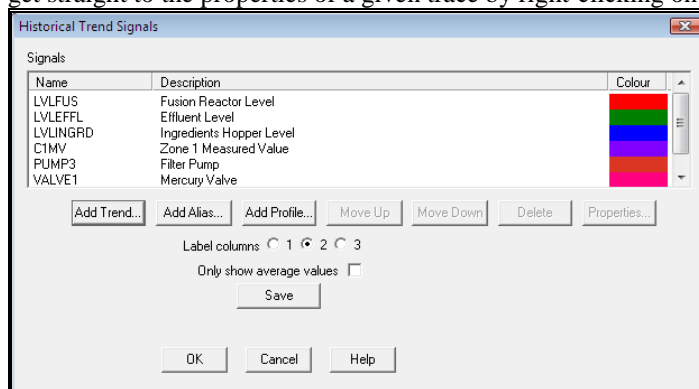
- **Time Span.**
Displays the dialog that allows you to alter the time span of the trend.



Select the required time scale. After selection, the time at the right hand of the x-axis will still be current time, but the left hand side will adjust to show a time that reflects the time span selected. The **Default** time span is that which was defined in the trend configuration, **Custom** is any value that you want, when any of the pre-defined scales will not suffice.

- **Trends.**

This displays a dialog showing all of the traces in the current trend. You can use this dialog to add and/or remove traces and to alter the properties of individual traces (alternatively you can get straight to the properties of a given trace by right-clicking on its y-axis).



The **Save** button on this dialog allows you to save any changes that you make to the trend so that they become permanent. Without this all changes will be lost when the trend is exited (i.e. you view a different Display). For the save button to be visible on the dialog the trend must be configured to allow run time saving (see Trends and Trend Properties in the Display Builder Section); you may optionally require a privilege in order to access the **Save** button.

- **No Graticule.**
This option removes all graticules from the display.
- **Restore Axis Scale.**
This option restores all y-axis scales to their pre-configured values, undoing the effect of all zooms and/or fit to graphs.

Historical Trending

As described earlier, a Trend object can be placed onto a Display Builder display. This object allows several signal values to be trended in real-time. As the signal value changes, its new value is plotted on the Trend display against the time at which the change occurred. Signals used in real-time Trends do not have to have any *recording strategy* associated with them.

When a Trend object is configured, one of its properties is to allow *Historical Trending*. If this option has been checked then the operator can double-click on the Trend object and it will expand to fill the entire screen. Its mode of operation changes from being *real-time* to *historical*.

In historical mode the data for each signal being trended is read from the permanent disk file. As such, signals displayed in a historical Trend must have a *recording strategy* associated with them. Any signals that were

defined for the real-time Trend that do not have a recording strategy will automatically be removed from the historical Trend.

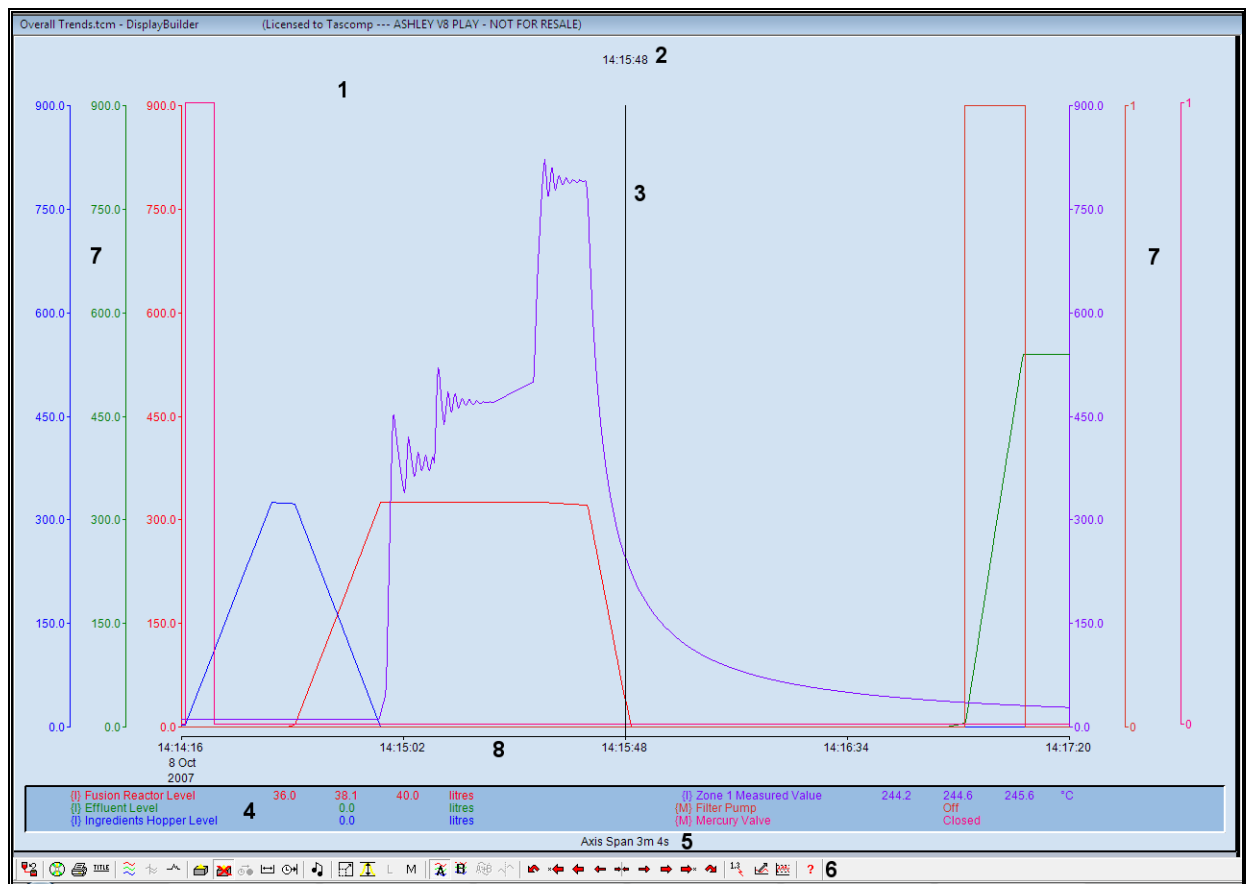
Archived Data

Historical trending allows archived data to be read from an alternative source, as well as the principal data area. Although primarily intended to be a CD-ROM, this could actually be any valid directory path. This is described in more detail later.

Alias Tags

Alias tags are used in association with batches and allow different signals to be shown on any one given trend depending on which batch is selected. If your trend uses alias tags, as soon as you switch to historical trending you will be prompted to select a batch.

The Parts of the Historical Trend Display



Graph Window (1).

All of the signals defined for this Trend will appear in this window.

Cursor Time (2).

The time of the current cursor position appears in this portion of the window, directly above the cursor. When viewing a batch in *Absolute Time* mode both the relative and absolute times will be shown. Also when in batch mode, the batch start and end positions are indicated by text that appears at this level of the display.

Cursor (3).

The cursor is positioned in time. Some of the toolbar buttons move the cursor left or right and the time at the current cursor position is shown directly above the cursor. The information window (4) shows the value of all of the signals being graphed at the current cursor time.

Information Window (4).

This contains the description of each signal being graphed along with the value of that signal at the current cursor time. There may be three different values shown for each signal, the **Min**, the **Mean** and the **Max**. If all three are the same then only one value is shown. For the reasons why these three values may be different, see **Average Mode**, below.

To the left of the signal description in braces { } is a code that gives additional information about the signal value at the current cursor time. The key to this code is:

- **A** - Alarm. The signal is in an alarm condition.
- **I** - Interpolated. The signal value is interpolated, i.e. the cursor lies at a point on the graph that is between two actual recorded values. The value at this point has been interpolated (linearly) between these two recorded values.
- **N** - Normal. The signal is an automatically recorded value, i.e. it comes from a piece of plant or was generated by some controls software.
- **M** - Manual. The signal is a manually entered value. This is most likely to have been done by using a slider or button on a Display Builder display or by entering values onto an XForm.
- **U** - Unknown. The value of the signal is unknown i.e. no value has been set for the signal, possibly because the plant equipment has failed or the communications with the plant has failed or because it is a manually set signal for which no value has yet been set.

Axis Span Information (5).

This shows the current axis span as a number of days, hours, minutes and seconds. When viewing a batch, this line also shows the current batch name, start time and end time.

Historical Trend Toolbar (6).

This gives access to all of the functions available within Historical Trending.

Y-Axes (7)

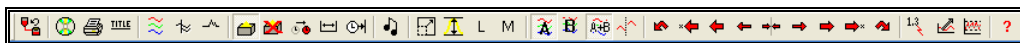
The y-axes are drawn at both sides of the trending area. You can right-click on any of the y-axes to display a menu of options, or you can shift-click and drag the axes as described in **Live Trending**, above.

X-Axis (8)

In historical mode the x-axis can also be dragged either left or right.

The Historical Trend Toolbar

When in historical mode the “**Historical Trend Toolbar**” is available to the operator. This controls a range of functions that are specific to historical trending.



The following features are available from the Historical Trend Toolbar (reading from left to right).

Back



This takes you back to the previous loaded Display. If you got into historical trending by double clicking on a real-time trend, it will take you back to that trend. If the historical trend is a permanent historical trend arrived at by clicking a button on another Display, it will take you back to that Display, and so on.

Data Location



As the quantity of recorded data increases it may prove beneficial, or indeed necessary, to archive earlier data off to some other location. The most suitable medium for this data is a recordable CD, removable hard drive or one of the “zip” style high capacity diskettes.

It is also possible that data may be archived to another computer. In this case, in order for it to be accessed by the Prodigy system the correct **Remote Access** parameters must be specified in the Prodigy Control Panel.

As mentioned earlier, historical Trends can take their data from some location other than the primary data location. Pressing the Data Location button on the Trend toolbar lets you choose the location of this secondary source. This will display a standard file requestor dialog, which is used to select the required data location.

***A Note on Archiving Data.** When using **Data Location** to select an alternative location in which to look for archived data, remember that the directory you select **MUST** be called **M00015180**. Where this directory resides is not important, but the name of it is. Bear this in mind when archiving data to some other media. Create a directory called M00015180 and then archive the sub-directories found within the M00015180 directory in the primary data area into this directory on the archive media. The following scheme is suggested.*

*1 On the alternative media create a directory with a sensible name, for instance **Jan 1998**.*

2 Below this create a subdirectory called M00015180.

3 Copy all of the subdirectories from the primary M00015180 directory that are dated Jan 1998 into the M00015180 directory within the Jan 1998 directory on the archive media (or use relevant CD recording software to achieve this end).

4 Delete the subdirectories that have been copied from the primary area.

*This way you will build up a CD with top level directories called Jan 1998, Feb 1998 etc. and the **Data Location** button can be used to browse into the relevant M00015180 directories below each year.*

Print

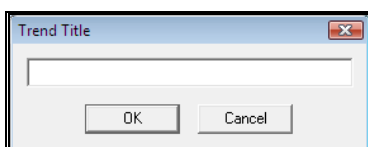


This brings up the Print Dialog to allow the current Trend to be printed.

Title



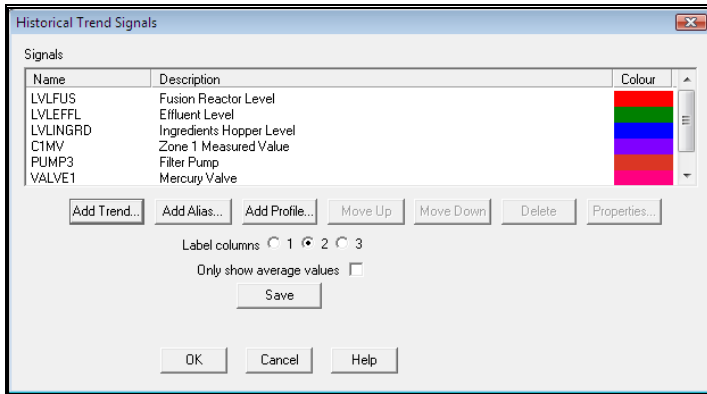
This allows a temporary title to be assigned to a Trend. This title appears underlined at the top of the Trend and will remain in place until historical Trend mode is exited. It is intended for labelling a trend prior to printing.



Select Signals



This displays a dialog showing all of the traces in the current trend. You can use this dialog to add and/or remove traces and to alter the properties of individual traces (alternatively you can get straight to the properties of a given trace by right-clicking on its y-axis).



The **Save** button on this dialog allows you to save any changes that you make to the trend so that they become permanent. Without this all changes will be lost when the trend is exited (i.e. you view a different Display). You may optionally require a privilege in order to access the **Save** button.

If the operator constantly has to alter the signals or properties of a signal in the historical Trend it may be wise to create a mini Trend with the correct signals and attributes defined, to ease the operators workload.

Align Curve



This displays a dialog that allows the user to align any or all of the Profile Curves present on the current Trend. Profile curves may be aligned to the left of the x-axis, to the batch start position or a position selected with the cursor.

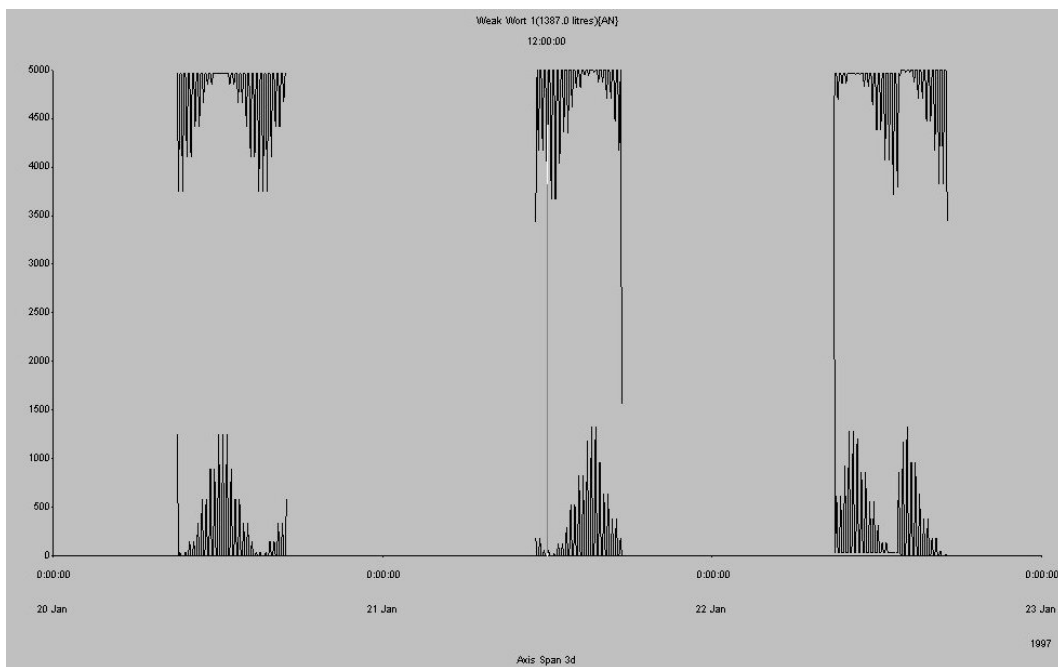
Peak Mode



When viewing data on a historical Trend it may be necessary for Prodigy to manipulate the data before displaying it. Take for instance the case where data was recorded every second for a given signal. If the area of the historical Trend window that is used to display the graph is 600 pixels wide and the timespan in use is 10 minutes then there will be one recording per pixel. If however a time span of 1 hour has been specified then there will be 3600 recordings to show on the graph, or 6 recordings per pixel. Prodigy will average the value of the recordings that occur at the same display position (pixel) in order to present the data in a meaningful manner – this is known as **Average Mode**.

As an alternative to Average Mode, data may be viewed in **Peak Mode**. This means that the highest and lowest values that occur at each pixel position are drawn as separate graphs. Obviously, if the time span and the recording interval are such that there is only one recording at each pixel position then there will be no difference between a graph drawn in Average Mode and one drawn in Peak Mode.

When this button is in the data is viewed in Peak Mode, when it is out it is in Average Mode.



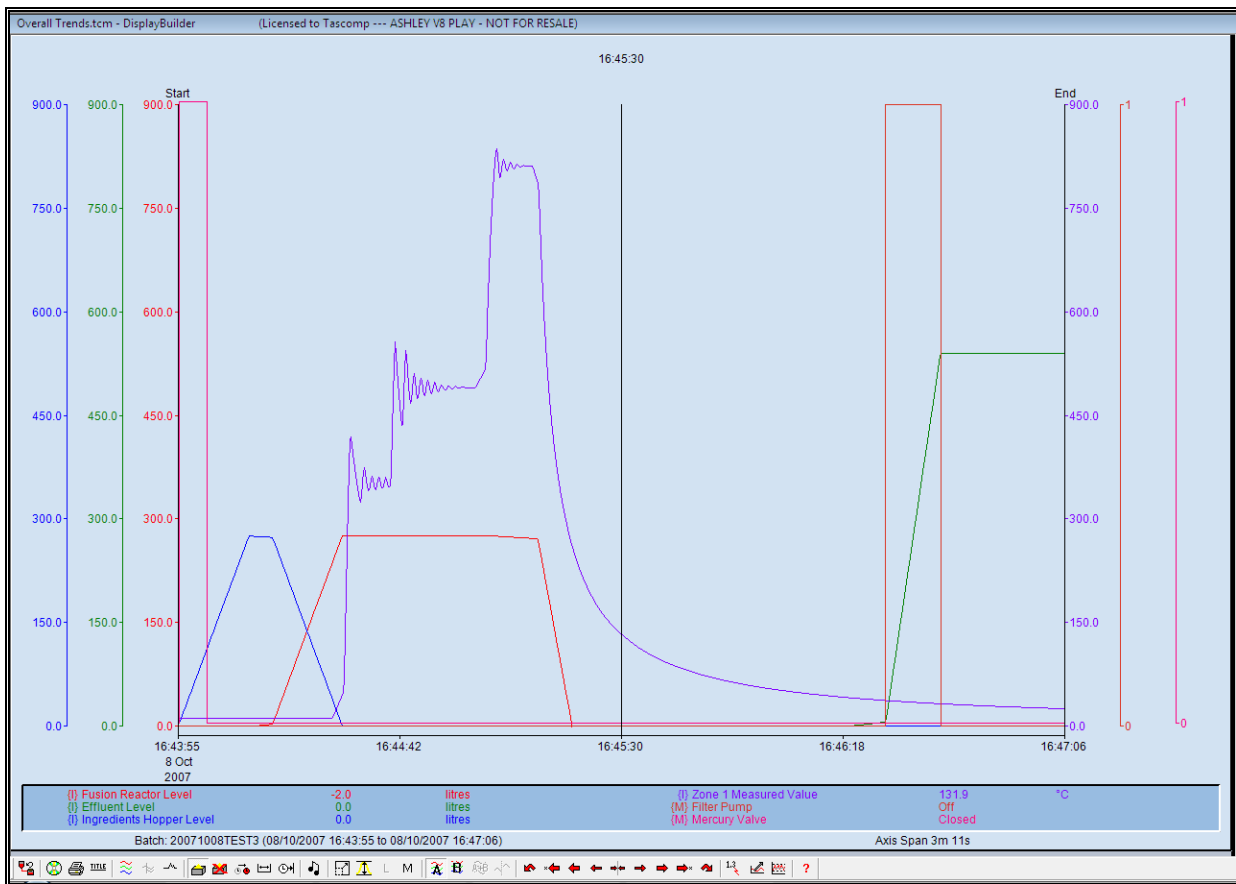
Select Batch



Prodigy allows the creation of **Batches**, each of which is given a unique name and which is delimited by time, i.e. a **batch start** and a **batch end** (see the relevant section under XForms). Clicking the **Select Batch** button presents the operator with a dialog that allows them to choose a batch. When a batch has been selected the following actions occur.

- The time span of the Trend will be adjusted to be the elapsed time of the batch.
- Any alias tags in the trend are de-referenced by looking up the alias in the selected batch record and extracting the real tag name to which it relates. This real tag is used in subsequent trend actions, until a new batch is selected.
- The relevant signal data for that time period will be displayed.
- The cursor is centred.

- A vertical line will mark the batch start time and a second line will mark the batch end time. These lines are both labelled.
- Information about the currently selected batch and its start and end times is displayed on the Trend.
- The Absolute Time and Relative Time buttons on the Historical Trend Toolbar become available.



The **Select Batch** button will remain indented and the **End Batch** button will become available.

End Batch



To exit from *Batch Viewing Mode* click on the **End Batch** button. All extra screen information relating to batches will be removed and the **Absolute Time** and **Relative Time** buttons will become unavailable

Relative Time



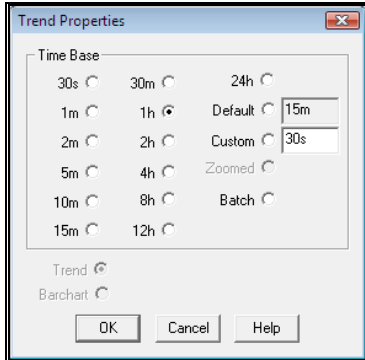
When viewing a batch the time (x-axis) may be displayed in one of two ways, **Absolute** or **Relative**. Absolute time is simply the time as it is normally displayed, e.g. 15:05:23 means 5 minutes and 23 seconds past 3pm on the given date.

Relative time means that time is displayed as the elapsed time since the batch started, so to use the same example as above, 15:05:23 means 15 hours 5 minutes and 23 seconds since the batch began. When operating in relative time mode the **cursor** is labelled with both the absolute time and the relative time of the current cursor position. Select **Relative Time** by clicking this button, revert to Absolute Time by clicking it again.

Time Span



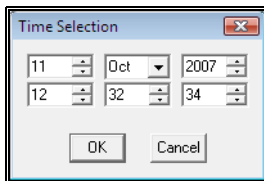
This brings up a dialog that enables you to select the time span that you want to use for the current trend. If you alter the time span, the graph will be centred on the current cursor position with the total time span of the X-axis being equal to that specified in the dialog.



Select Date/Time



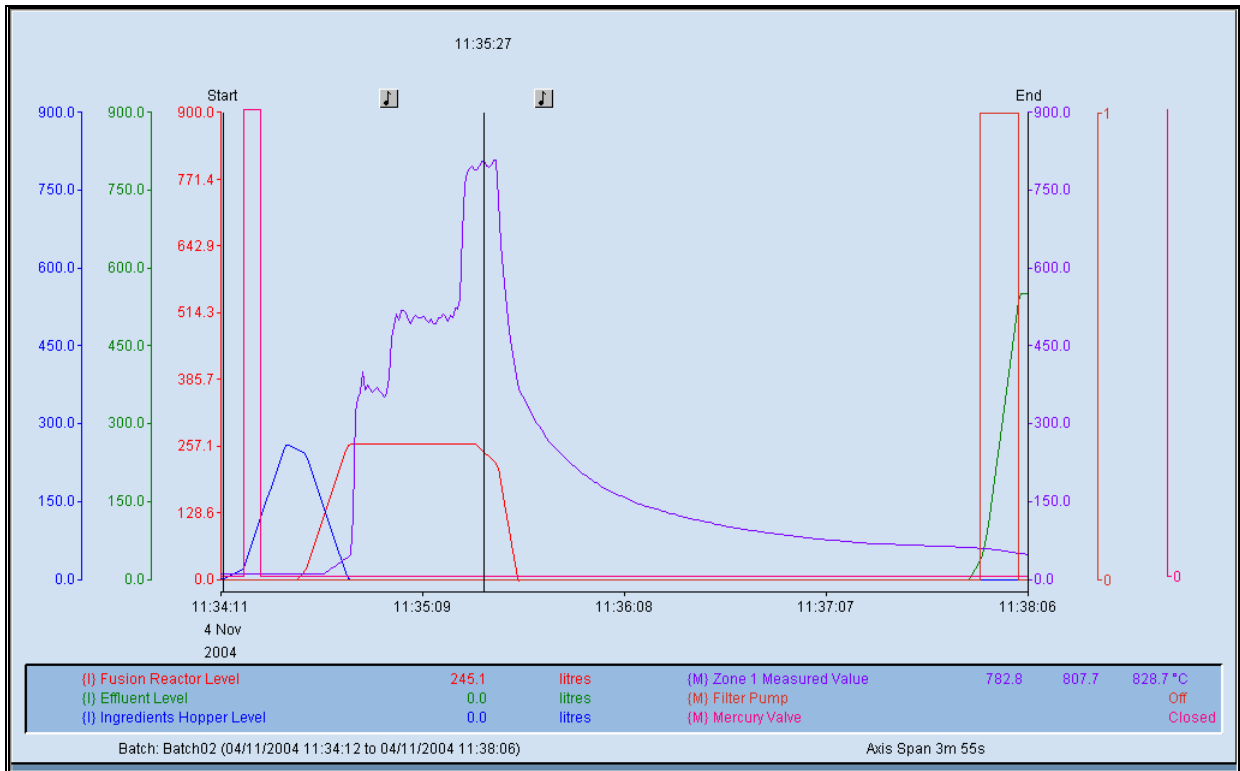
This button allows you to select the date and time that will be shown at the right hand side of the x-axis. The date/time at the left-hand side will be calculated according to the current time span.



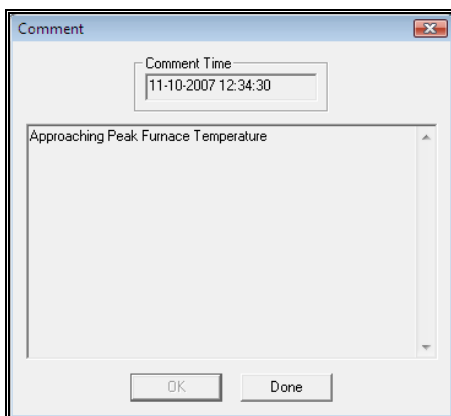
Comments (Notes)



During the process cycle the operators may be required to make certain observations or take manual readings. It could be that this forms a vital part of the QA procedure. Prodigy provides a commenting facility that allows any observations to be entered and timestamped. When viewing historical Trends the positions of any comments may be overlaid on the graph by pressing the **Comments** button on the Historical Trend Toolbar. Any comments that have been entered for the time period currently being viewed will have their positions marked by a note icon.



Clicking on one of these icons will present a dialog that displays the comment text.



Zooming

Zooming in is not accessed via any of the toolbar buttons but by dragging a box on the graph over the area that you wish to zoom in on. As the box is dragged an outline is drawn to indicate the selected area. When you release the mouse button a dialog is presented informing you of the time period over which you have zoomed and asking for confirmation. If confirmation is given the graph is redrawn with the zoomed-in time frame.

Zooming in affects both the time frame (x-axis) and the signal value (y-axis) so take care to ensure that the required range of the signal is included in the drag box. If you happen not to, this can be rectified by using the "Undo Zoom" button, described later.

Undo Zoom



This cancels any zoom operation that is in effect and restores the signal scale and time axis to their full values.

Fit to Axes



This option causes the system to look at the graphs that are currently visible and re-scale each y-axis so that it runs from the minimum value that is currently visible to the maximum, allowing for a margin at the top and bottom. This means that a signal may now use more space in the vertical axis than it had been previously, exaggerating any changes in signal value.

Two side effects can occur that may confuse the operator.

Firstly, if for instance one graph was a flat line at say 80 and another was a flat line at 55, when axis expansion occurs the graphs will be co-incident with each other. The operator may be confused into thinking that one graph has "vanished".

Secondly, the relative scale between the axes of the graphs being viewed is lost. A signal that varied between 1 and 10 on an axis of 1 to 100 will now appear on an axis of around 1 to 10. A signal that varied between 1 and 100 will appear unchanged, but the two signals will now use the same amount of y-axis. Be sure to read off from the correct axis.

Unclick the button to undo the effect of **Expand Vertical Axes**. All y-axes are restored to their full values.

Less Smoothing



This is simply the reverse of the procedure described below, the amount of smoothing being successively reduced until the original granularity is restored.

More Smoothing



Sometimes a signal may fluctuate so much that the underlying Trend is difficult to see. Prodigy trending provides a facility to smooth such signals as they are viewed. Successive levels of smoothing remove more of the minor peaks and troughs (noise), allowing the underlying shape of the trend to show through clearly.

Multi Trending

A specialised facility within Historical Trending is **Multi Trending** and it is controlled by the next set of buttons on the Historical Trend Toolbar. Multi Trending allows two different sets of data for the same set of signals to be displayed and compared against each other. One set of trends is shown in full-brightness (foreground) while the other set is dimmed (background). You can toggle which set is which by means of the "A" and "B" Trend buttons. The axes (all y-axes and the x-axis) and the information area show details of the current foreground trend.

The two sets of trends can be displayed “on top” of each other or side-by-side and all axes for either set of trends can be dragged, as described later.

In its easiest-to-use form it allows two different batch runs to be compared. An example of how to do this is:

- Press the Batch button to select the first batch you are interested in.



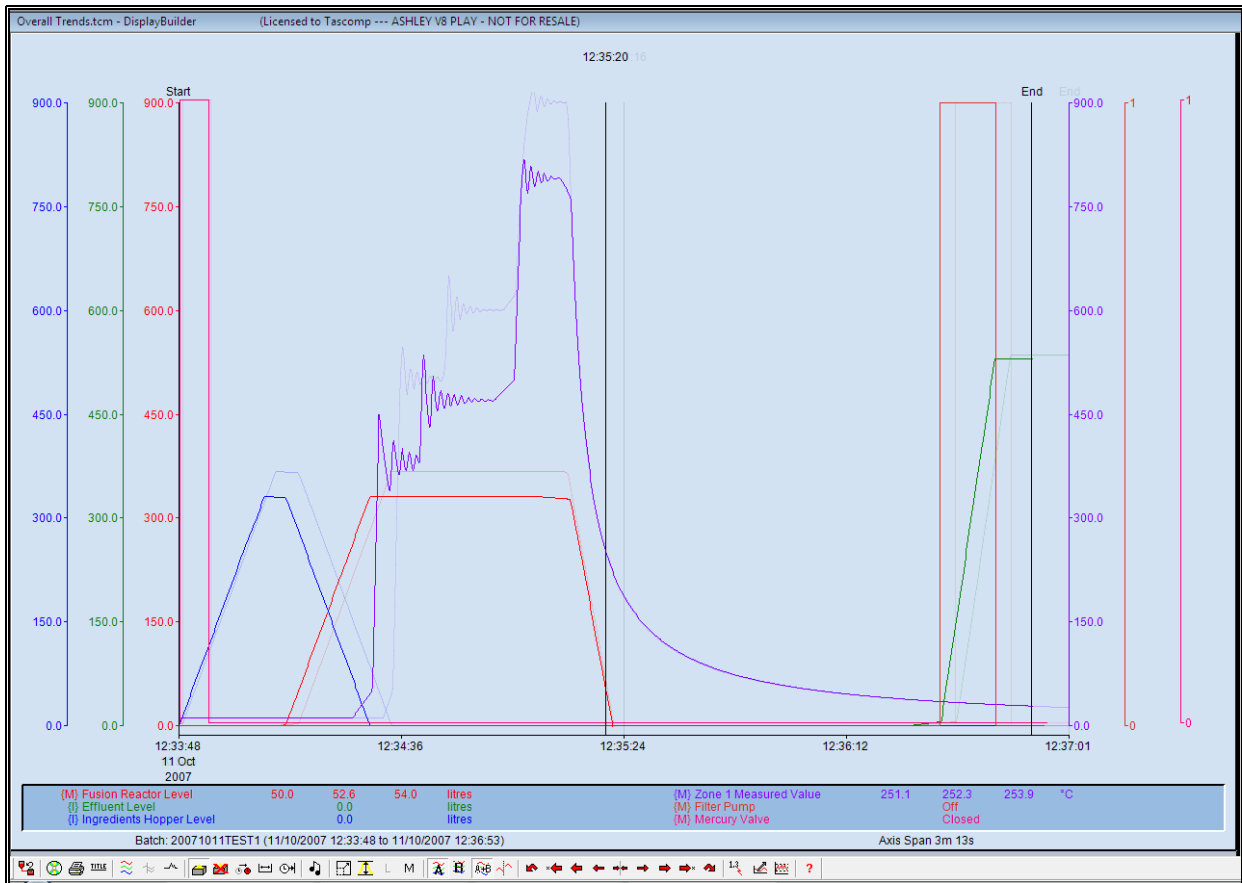
- Press the **Show “B” Trend** button.
- Select the second batch that you are interested in.
- Switch which set of trends are shown as foreground by toggling between the A and B trends (click the relevant button).



Show “A” Trend



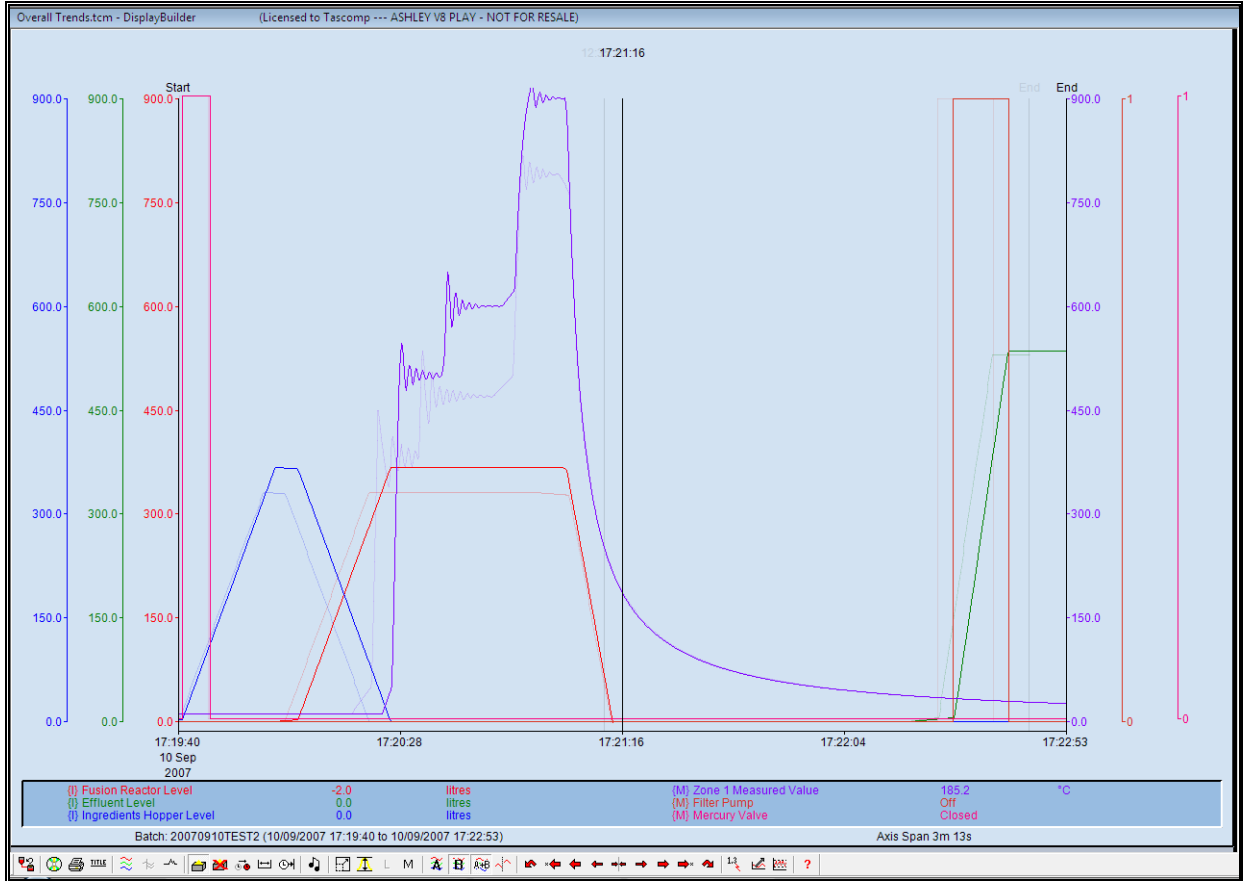
Makes the “A” trend the foreground trend.



Show "B" Trend



Makes the "B" trend the foreground trend.



Link Trends

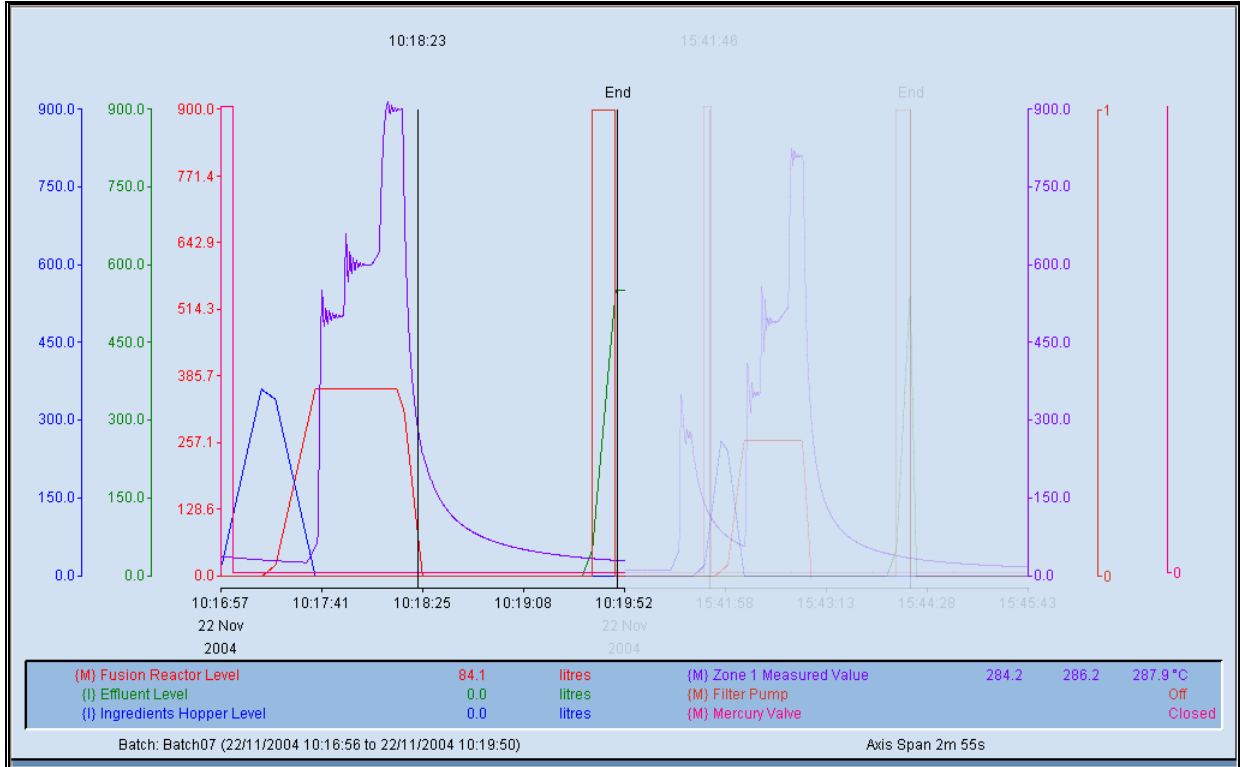


When this button is depressed (and that is the default state) then anything that is done to one trend (i.e. A or B) is also applied to the other. For instance, if you have A trend as the foreground and you zoom in or drag one of the axes it will have the equivalent effect on the B trend. Unclicking this button allows you to act on one set of trends without affecting the other.

Split Trend Display



As well as being able to display the A and B trends over each other there is also the option to display them side by side. Click this button to switch the display to the style shown below:



Previous Period



This button has the effect of scrolling back by one complete time span.

Previous Value



When Prodigy draws a graph there are an arbitrary number of points per graph which are joined by straight lines. As you move the cursor over a graph the value at the current point is displayed in the information window. It is possible that the value shown has been interpolated and is not an actual recorded value. Pressing the **Previous Value** button causes the cursor to move to the next value to the left of the current position that is an actual recorded value. Note that this may be a value from any one of the graphs being displayed. Look at the information window to see what is going on.

Large Left Scroll



Move the cursor to the left by an amount equivalent to $1/25^{\text{th}}$ of the current axis span.

Small Left Scroll



Move the cursor to the left by an amount equivalent to $1/300^{\text{th}}$ of the current axis span.

Centre Cursor



Redraw the display so that the current cursor position is moved to the centre of the screen. The time at the left of the x-axis will be *current cursor time - (axis span / 2)*; The time at the right of the x-axis will be *current cursor time + (axis span / 2)*.

Small Right Scroll/Large Right Scroll/Next Value/Next Period

These are as for the options shown above, but for “next” rather than “previous”.

Get New Values



If the portion of the graph that is being viewed shows the current time, it is possible that new values have been recorded but are not currently shown on the graph (because the graph was drawn before these values were recorded). Pressing this button brings any newly recorded values into the graph.

CSV Data Export



You may wish to export the data being viewed so that it can be loaded into some other package, e.g. a spreadsheet. The **CSV Data Export** facility makes this possible. Clicking the button displays a dialog that asks for some extra information and control parameters.

- **Download to profile curve.**

This indicates that the selected curve is to be downloaded to a CSV file that will be used as a profile

curve. It automatically selects the correct options for **Labels** and **Time Axis** and sets a different extension type for the **Filename**.

- **Signals.**
This is a list of the signals currently being trended. All of the signals in this list will be exported. If you do not want to export all of these signals, they can be removed by pressing the delete button and in the same way extra signals can be exported by pressing the add button and choosing the appropriate signals required.
- **Comment.**
Any text entered here will be added to the CSV file so that it may be used as a heading when imported or simply for reference.
- **Filename.**
The file that the CSV data will be placed into. The **Browse** button can be used to select the required path.
- **Start.**
This is the start time for data export. It defaults to the current start time of the Trend being viewed. Altering this field will affect the **End** which is calculated to be **Start + Duration**.
- **Duration.**
This defaults to the axis span of the current Trend. Altering this field will affect the **End** which is calculated to be **Start + Duration**, and may also have an effect on the **Slices** and **Interval** fields.
- **End.**
This is the end time for data export. It defaults to the end time of the current Trend. Altering this field will affect the **Duration** that is calculated to be **End - Start**, and may also have an effect on the **Slices** and **Interval** fields.
- **Use Actual Recordings.**
If selected all the data points relating to the signals that are selected is sent to the CSV file. This means that the number of data points will relate to the signal that has the highest sample rate and signals with lower sample rates will have the intermediate points calculated using interpolation. Data points are included whenever any of the signals selected changes.
- **Slices.**
This is the number of data points that are required and is by default the number used to produce the graph as seen on the screen. This means that if any smoothing has been applied the number of slices will be less than if no smoothing was applied. If no smoothing has been applied, the number of slices can be reduced manually to create a smoothed graph, though obviously this is best done via the **M & L** buttons so that the results can be viewed prior to performing a data export. The number of data points and the duration combine to give the **Interval** (described next), the **number of slices * the interval** being equivalent to the **duration**.
- **Interval.**
This is the time between successive data points or **Slices** and altering this value causes the Slices value to be recalculated, based on the **Interval** and the **Duration**.
It is really the number of Slices that determines how the data is generated. The Interval field gives a more convenient method of determining the number of slices by allowing the user to specify the Duration of the graph and the time required between slices.
- **Launch Excel.**
Launches Excel™ with the CSV file produced by the data export.
Excel should be installed on your system and should be associated with the file type .xls. If Prodigy cannot find Excel it will prompt you to locate it manually by presenting a browser window. Once located its location will be stored for future use.
- **Labels.**
Determines whether or not the **Comment** entered on this dialog, along with some other informational text (Tag Names, Descriptions and Units), is output to the CSV file. This text may be used as headings for graphs etc. produced by the spreadsheet created from the CSV data.
- **Graph.**
Determines which of the data points to export, the **Min, Mean** or **Max**.
- **Data In.**
Determines whether the data is exported with times running in rows and their respective values beneath them or with times running in columns and their respective values to the right of them.

The **Relative to start** check box means that times are also output in another column as seconds (or fractions of seconds) from the start time. In other word, the time output is **time of slice - time of first slice**. The **Fractional Time Column** check box includes an additional column that shows at what point during a specific second in time a piece of data was obtained for a signal.

SPC



Pressing this button will launch an SPC analysis of the signal currently being viewed on the historical trend (other than a digital signal). There should only be one signal trace on the trend and a suitable SPC configuration should have been created for analysis of this signal.

See the section on SPC for more details.

Help



This displays help on the Historical Trend Toolbar.

Zooming, Dragging Etc.

You can zoom in on trends and drag the axes in historical mode just like you can in real-time mode.

Data Value Read Off

The information window shows the values of all of the signals on this Trend at the given cursor position. The cursor may be moved left or right in a number of ways as described earlier. The cursor may also be positioned directly by clicking in the main graph window at the point at which the cursor is required.

Remember that the mnemonic code to the left of the signal description in the information window gives information as to whether the signal value being read off is real or interpolated, in alarm etc. (see The Parts Of The Historical Trend Display).

Digital and Finite State Signals

Because of their discrete nature digital signals (and to a lesser extent finite state signals) will often obscure each other. If you have three digitals you will be able to see that one is at value 1, one is at value 0 and you will have no way of seeing the value of the third, as it will coincide with one of the others. To eliminate this problem, each digital and finite state signal is shifted a number of pixels vertically so that it does not coincide with other signals. The axes of these signals are also shifted vertically to maintain the correct Y-axis intercept positions. Bear this in mind when viewing Trends.

Menu Bar

What Is the Menu Bar?

Access to the Prodigy suite of programs is via a *Menu Bar*. This may be automatically invoked at system start up (see **Display Launcher Configuration**) and will appear at the top of your screen.

The Prodigy Menu Bar consists of a toolbar containing a number of buttons. Each button may invoke a command or provide access to a menu. Each option on these menus may invoke a command or may provide access to further menus.

Additionally, the Prodigy menu bar allows users to log on and off in systems where Prodigy Security has been implemented. The menu bar is also capable of showing the system clock.

The default Prodigy Menu Bar that is created when Prodigy is installed contains a standard set of options. An example of a Menu Bar is shown below.



Colour is an important aspect of the Menu Bar and should be used to make it both visually appealing and easy to use.

To customise the menu bar right click and select properties.

Customising the Prodigy Menu Bar

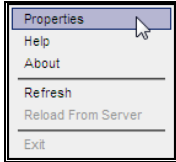
The Prodigy Menu Bar behaves in much the same way as the standard Windows Task Bar.

- You can specify which programs it gives access to.
- You can display program icons next to the items in the menu.
- You can vary the format of menu bar items and the colour of the menu bar.

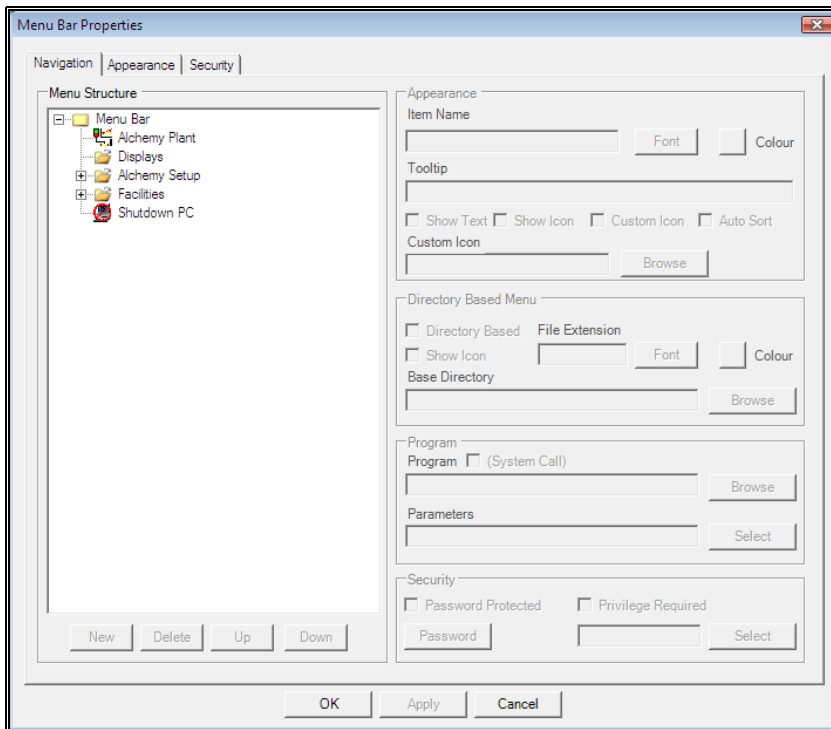
The Navigation Tab

The purpose of the Menu Bar is to enable the user to have access to the Prodigy suite of programs and also to any other programs on the PC, as the system manager sees fit.

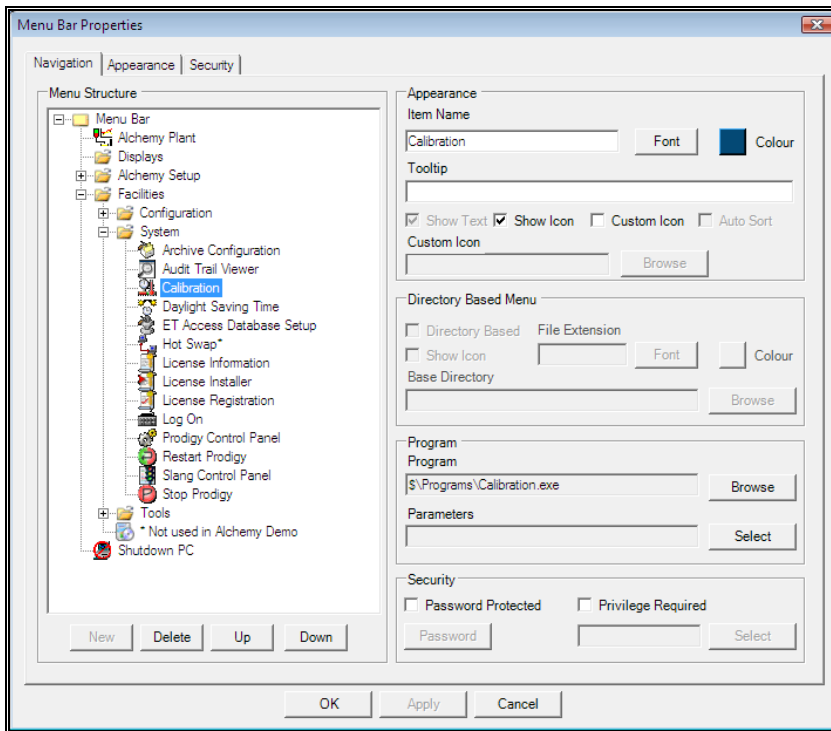
To configure the Menu Bar right-click on it on the grey area that is not one of the buttons. Select **Properties** from the menu.



This will display the **Menu Bar Properties** window that has three tabs; Navigation, Appearance and Security.



The buttons and menus are configured by using the **Menu Navigation** tab, as shown below.



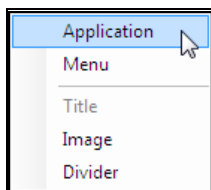
This tab on the properties dialog allows the user to define which programs can be accessed from the Menu Bar and how the menu structure is laid out.

The dialog has two main sections to it. At the left hand side of the dialog is a *tree control*, which is a graphical representation of the menu structure. Each branch on the tree can be *expanded* or *collapsed*. When expanded all sub-branches can be seen; when collapsed only the branch itself can be seen.

Adding Menu Items

To add items to the menu follow these steps:

- First expand the tree to the extent that you can see the point at which you want to add the new item.
- Select the menu item which you would like your new item to be pasted onto.
- Press the **New** button at the bottom of the tree structure section (if the menu item you clicked on is either a Program or a Directory Based Menu then New will not be available).



Select the Menu Item that you want to add to the menu bar tree structure (Items can also be added by right clicking on the appropriate point in the tree and selecting new).

The new item will be inserted into the tree control and can be moved up or down by pressing the **UP** or **Down** buttons respectively.

Titles can be added, these are items that cannot be clicked and they can be used to indicate what area of Prodigy a group of programs belong to.

Images can be added to certain menus, one example of the use of this item could be if a company wants their logo on the main menu bar.

Divider items can be inserted, these can again be used to separate and group similar items together.

Defining Properties for Menu Items

At the right hand side of the dialog is the area where the actual properties of items in the menu are entered. Some of the properties are only relevant for **menus** (or **directory-based menus**) and some for **applications**.

Menu Item Properties

The screenshot shows a dialog box titled "Menu Item Properties" with the following sections:

- Appearance:** Item Name (Facilities), Font button, Colour (red box), Tooltip field, Show Text (checked), Show Icon (checked), Custom Icon (unchecked), Auto Sort (unchecked), Custom Icon field, Browse button.
- Directory Based Menu:** Directory Based (unchecked), Show Icon (unchecked), File Extension field, Base Directory field, Browse button.
- Program:** Program field, Browse button, Parameters field, Select button.
- Security:** Password Protected (unchecked), Privilege Required (unchecked), Password field, Select button.

- **Item Name.**
The name that will appear on the menu.
- **Tool Tip.**
Enter text here that will appear as a tool tip when the user hovers the mouse cursor over a menu option.
- **Colour.**
To change the colour of an items text, click on the coloured box to the right of the Item Name. The colour can be set for a menu item, and if so the option is given to use that colour for all sub-menus and items within those menus. Alternatively the individual menu item (or application) may be given a specific colour.
- **Font.**
Click this button to select the font to be used for this menu item.
- **Show Text.**
Checking this box makes the item name as entered above appear on the menu bar.
- **Show Icon.**
Checking this box causes an icon associated with the program that will run to be displayed on the menu bar (if available).
- **Custom Icon.**
Checking this box causes an icon that you have selected through the Custom Icon **Browse** button to be displayed on the menu bar.
- **Auto Sort.**
Checking this box makes the menu bar sort all of the items alphabetically on a drop down menu.
- **Directory Based Menu.**
See later.
- **Program.**
See later.
- **Security.**

Password.

Checking this box means that the menu item will only be accessible if a valid password is entered. Pressing the **Password...** button allows you to specify the password that must be entered. A dialog will be displayed that takes the password along with a confirmation of the password to ensure correct typing.

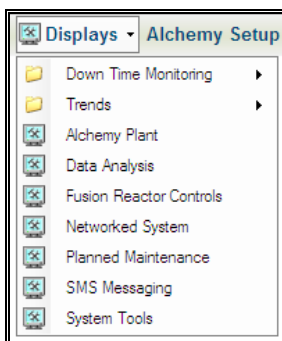
Privilege.

Checking this box means that the menu will only be accessible to users who have the required privilege. Click the **Browse...** button to select the required privilege.
(Password & Privilege are mutually exclusive).

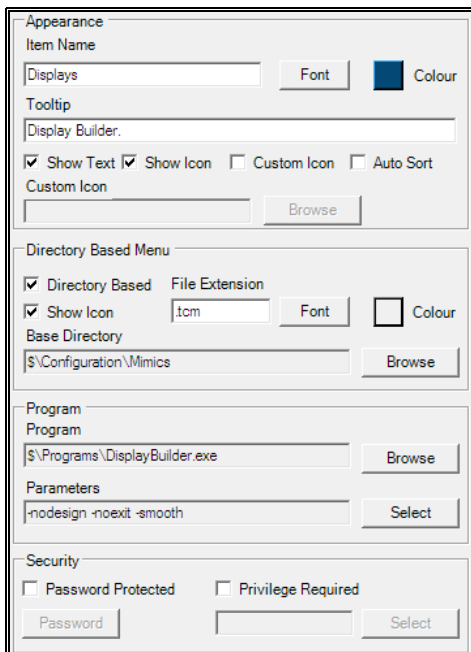
Directory Based Menu Item Properties.

These are self-generating menus that scan a given directory for files with a given extension. A menu is generated containing each item that matches the given extension – the name that appears in the menu will be the name of the file minus the extension. Subdirectories are also shown on the menu along with any matching files within the subdirectories (see example below).

The directory that is selected is constantly scanned for changes (new files, deleted files, renamed files etc.) and any changes are instantly reflected in the menu.

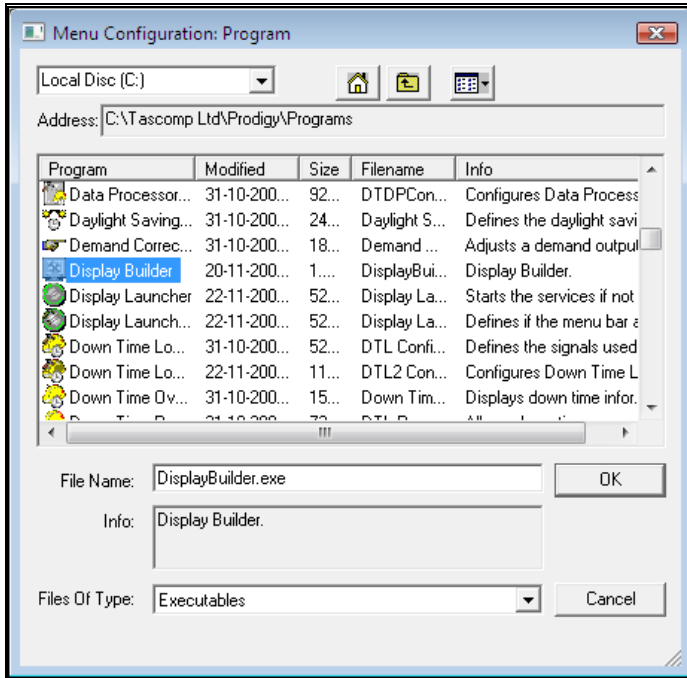


Configure the Directory Based menu as shown below:

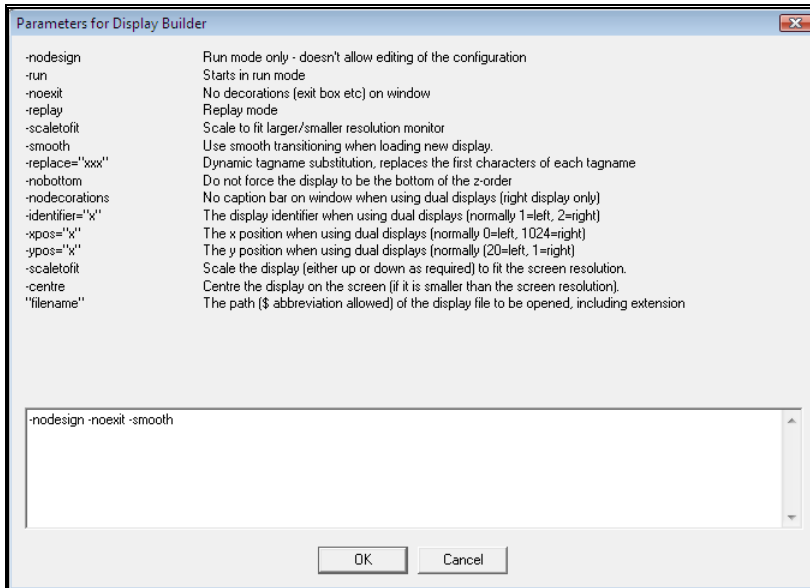


- Check the **Directory Based** box.
- Enter a **File Extension**.
For Prodigy this will most likely be .tcm (**Display Builder** definition files), though you could specify files that other programs recognise, for instance .doc for WinWord or .txt for Notepad
- Select a **Base Directory**.

- Select the **Program** to be used.



- Give any additional **parameters** the program needs.

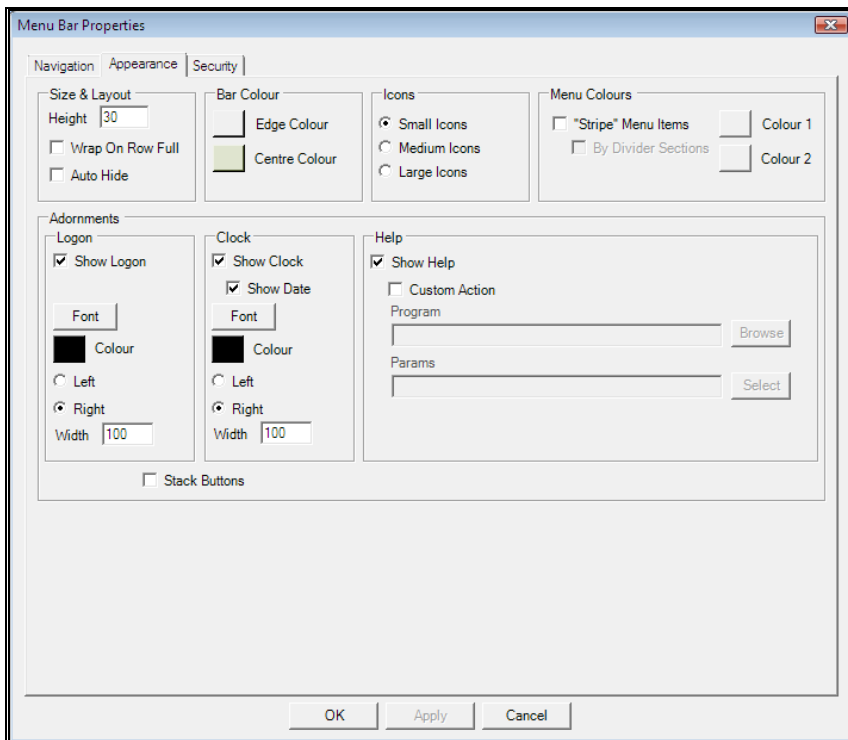


Non-Prodigy programs and windows utilities, for instance notepad, may understand certain parameters (e.g. /p for print and so on). Find these parameters from the relevant documentation for those applications and enter them manually in the Parameters dialog

A word on subdirectories

It makes good sense to arrange your Displays into subdirectories if there are several Displays in the system. These may be grouped by plant area or function, for instance. Directory-based menus look for subdirectories and make matching submenus for these.

The Appearance Tab



- **Size and Layout.**

The value in the **Height** box determines the height of the menu bar, the value entered is the number of pixels that will be taken up vertically across the screen. If the **Wrap On Row Full** box is checked the menu bar options will start on a new line if there are so many that they reach the right side of the bar where logon, clock and help are located see below (the menu bar must be tall enough to accommodate two rows of text).



Checking the **Auto Hide** box turns the Auto hide function on that causes the menu bar to hide off the top of the screen when it is not in use, this can save space for displays as detailed earlier.

- **Bar Colour.**

This section allows the colour of the menu bar to be changed; the edge colour and centre colour can be picked and then the bar will be shaded between these two colours. An example of a shaded menu bar from yellow to orange to yellow can be seen below.

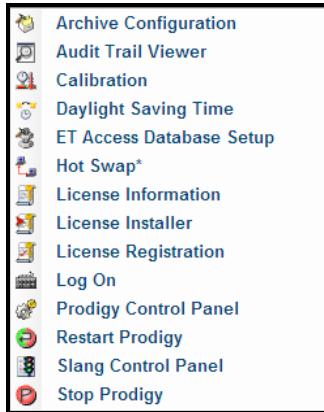


- **Icons.**

If the **Show Icon** check box (detailed earlier) is checked then this option applies; it controls the size of the icons that will be displayed on the menu bar, there are three sizes each is detailed below.

Small Icons.

Small icons are displayed. The Menu is as below.



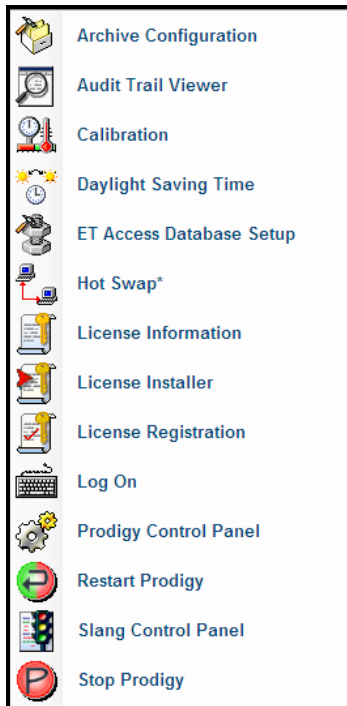
Medium Icons.

Medium icons are displayed. Note that these are interpolated from the large icons and so may not appear “perfect”.

Large Icons.

Large icons are displayed. This gives the best results visually but menus take up more screen space and it may not be appropriate for menus with many entries.

The Menu is as below.

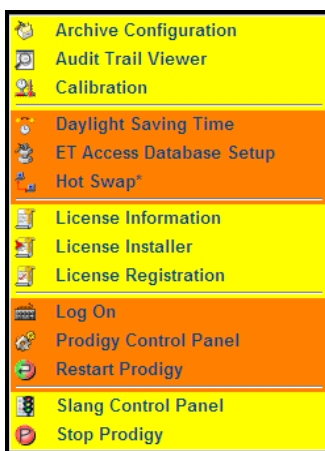


- **Menu Colours.**

Checking the “**Stripe**” **Menu Options** box allows two colours to be selected that will be used as background colours for the drop down menus. The effect of this can be seen below where yellow and orange have been selected.



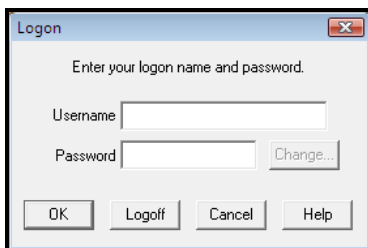
If the **By Divider Sections** box is checked the menu will be shaded similar to above but rather than per line it will be by divider sections, this can be seen below.



If the menu options need to have a single background colour then select **“Stripe” Menu Options** and pick the same colour for both of the options.

- **Logon.**

Checking the **Show Logon** option causes a button-like section showing the currently logged on user to be displayed on the menu bar. The font and colour can be edited by clicking on the relevant options and the logon can be positioned at either the right or left of the menu bar. The width of the Logon section can also be specified, this is the number of pixels that it will take up in the horizontal direction. Clicking in the area of the logon section brings up the following dialog that allows a different user to logon, the current user to log off or a user to change their password.



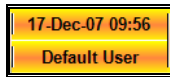
- **Clock.**

Checking the **Show Clock** option causes the current time to be displayed at on the toolbar. By checking the **Show Date** item the date and time will be shown. The time is always shown as a 24-hour clock and the date is always of the form 24-Aug-07. The font, colour, position and width can be edited in the same way as the **Logon** section.

- **Stack Buttons.**

Checking this option causes the Logon and Clock to be placed above and below each other as can be

seen below. It is important that the menu bar is tall enough to accommodate the heights of both the logon and the clock



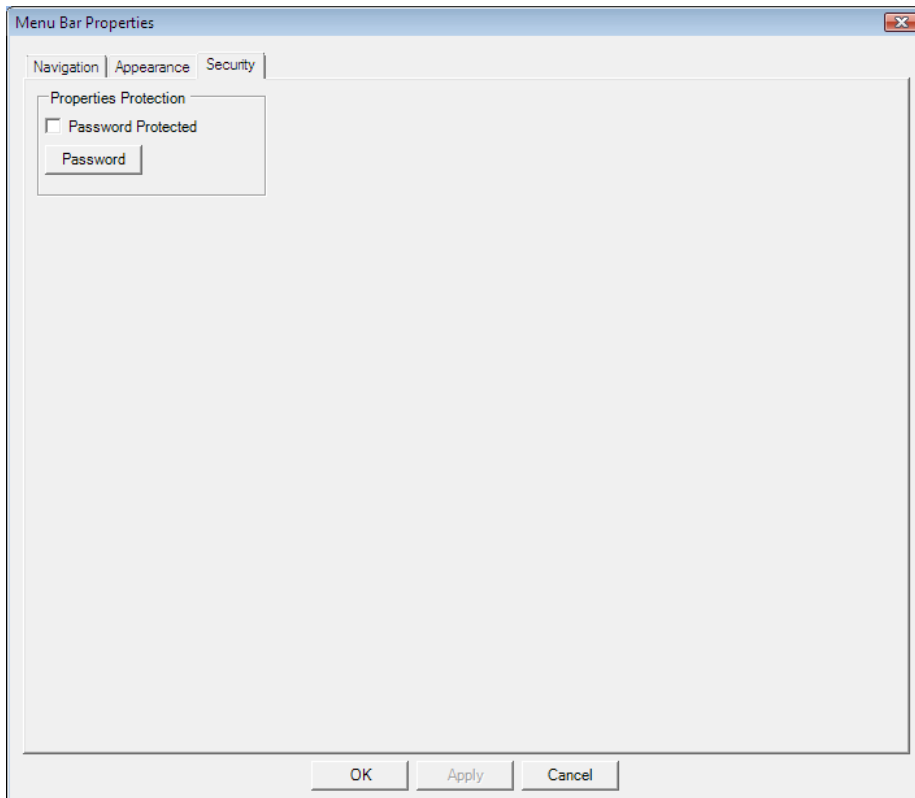
- **Help.**

Checking the **Show Help** option causes the icon that can be seen below to be placed at the right hand side of the menu bar.



If only the **Show Help** box is checked clicking the icon will cause the Prodigy help file to be loaded with the information from this section of the manual relating to the menu bar to be loaded. Selecting **Custom Action** allows you to make the button do a custom feature; an example use may be to load up an XForm, Display or document with some generic instructions for a user.

The Security Tab



Properties Protection

In order to implement a more secure system it is probable that some of the Menu Bar buttons or commands will be password protected, as described in **Defining Properties For Menu Items**. To prevent operators removing this password protection and also to stop any other unwanted tampering with the commands available on the Menu Bar it will be necessary to protect the *Properties...* menu item. Select the **Tool Bar** tab on the Properties dialog. This has a check box for **Password Protected**. Check this box and press the **Password** button. Enter a password into the password dialog that appears.

If you are implementing a Users based system, you should not use this option. In a Users based system the privilege "Configure Menu Bar" is required to allow access to the Menu Bar properties dialog.

As with all password-protected systems, care should be taken over the password chosen. Making it too obvious will defeat the object. Try to use a mix of upper and lower case

characters and numbers. If you forget the password it is not easy to remedy the situation - you must contact your supplier.

Impact of Security on Menu Bar Appearance

If you have applied security to any of the menu items (either passwords or privileges) then the protected items will appear greyed out.

If the logged-on user has the required privilege then the item will not be greyed out but will appear in full colour. Every time a different user logs in or out the menu is re-evaluated to take account of the privileges that the new logged in user has.

If a menu item is greyed out it does not mean that the user cannot access that item, just that they will be prompted for a user ID/password when they select it.



Alarms

What Is an Alarm?

An alarm condition arises when a signal value goes beyond some limit that defines the boundary of normal operation. What these boundaries are and whether or not exceeding them should trigger an alarm will depend upon just what it is that the signal is monitoring. In Prodigy any number of signals can be designated as having alarm conditions and up to four conditions per signal may be defined. These conditions represent **High**, **Very High**, **Low** and **Very Low** and are defined in the **Signal Database** (see **Alarms in Signal Database Configuration**). How to do this is described in the relevant section.

The Alarm Display

When a signal goes into an alarm condition a message is placed in the **Alarm Banner**. This is a toolbar type of dialog, much as the **Menu Bar** described above is. The Alarm Banner is docked at the top of the screen.



Right clicking on the alarm banner displays the context menu that allows you to define the banners properties. The alarm banner is **always on top** of the screen and occupies an area defined in the properties.

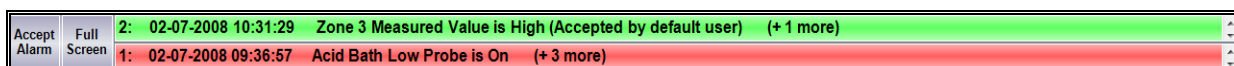
What Appears In the Alarm Display

The Alarm Display has two modes, **Banner** and **Full Screen**.

In Banner mode (shown below) the Alarm Display shows the **oldest unaccepted alarm** at the bottom of a two-line window and the **most recently accepted alarm** on the line above this. If there is more than one unaccepted alarm the alarm message is followed by some text that indicates how many other alarms exist.

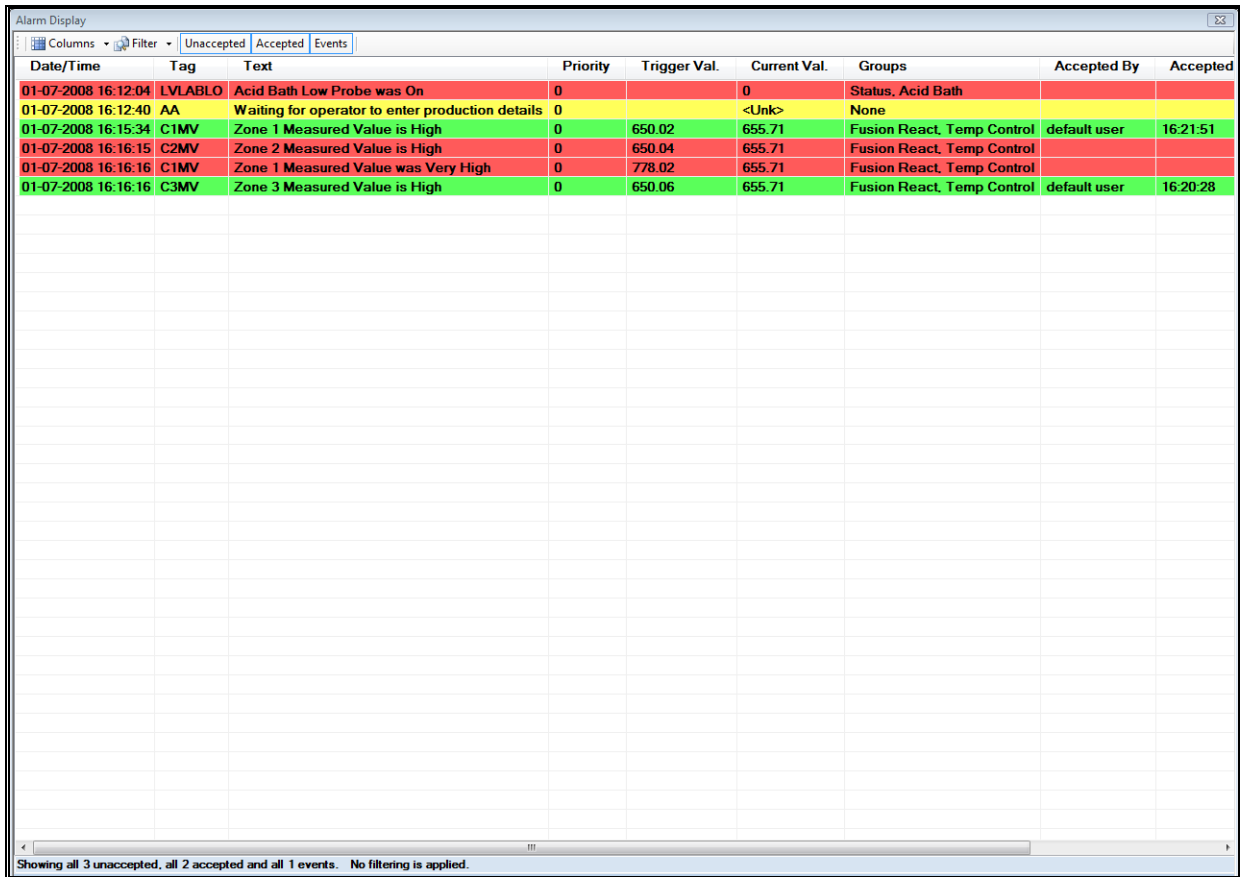
To the right of the two line window there are two up and down arrows, these can be used to cycle through the additional alarms that are displayed, this allows you to observe other signals that are in alarm or events without having to accept alarms or move into the **Full Screen** view. The number to the left of the two line window indicates the alarms position in of the alarm that is currently been viewed i.e. in the screen shot below the top line is showing the second alarm that was accepted.

*If you are using alarm priorities the bottom line of the display show the **oldest unaccepted alarm of the highest priority**.*



When you create alarm strategies you define the text that is to be displayed as part of the alarm message. In the example above this text was **High** and **On**. While the alarm condition is true the message reads **is High**, but if the value goes out of alarm without being accepted this text changes to read **was High**. This means that the operator can see that the value was in alarm at some point, even if it is not now.

In Full Screen, the Alarm Display shows the current state of all signals that are in alarm.



Date/Time	Tag	Text	Priority	Trigger Val.	Current Val.	Groups	Accepted By	Accepted
01-07-2008 16:12:04	LVLABLO	Acid Bath Low Probe was On	0		0	Status, Acid Bath		
01-07-2008 16:12:40	AA	Waiting for operator to enter production details	0		<Unk>	None		
01-07-2008 16:15:34	C1MV	Zone 1 Measured Value is High	0	650.02	655.71	Fusion React, Temp Control	default user	16:21:51
01-07-2008 16:16:15	C2MV	Zone 2 Measured Value is High	0	650.04	655.71	Fusion React, Temp Control		
01-07-2008 16:16:16	C1MV	Zone 1 Measured Value was Very High	0	778.02	655.71	Fusion React, Temp Control		
01-07-2008 16:16:16	C3MV	Zone 3 Measured Value is High	0	650.06	655.71	Fusion React, Temp Control	default user	16:20:28

In addition to alarms *Events* can also appear on the Alarm Display. Event messages are generated by the system (most usually by a SLANG program) in order to pass information to the operators, or to make them aware of something. Some events may automatically remove themselves from the Alarm Display after a few seconds.

Accepting Alarms

The operator can *Accept* an alarm. This informs the system that they are aware of the condition. It does not necessarily mean that the cause of the condition has been rectified. In the example shown above the accepted alarm reads “*Zone 1 Measured Value is High*” – it is **still** high even though it is accepted.

Accepting an alarm removes the message from the display (it will become the most recently accepted alarm) and stops the warning sound (provided they are configured and that other alarm conditions do not exist). If the cause of the alarm condition has not been rectified no further warning will be given in the Alarm Banner unless Alarm Nag has been specified (see below). The warning will still be shown when in Full Screen mode.

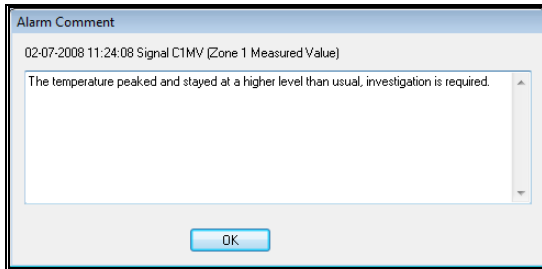
An alarm can be accepted by either double clicking on the alarm displayed on the banner or pressing the accept alarm button to the right of the alarm banner.

Alarm Nag

If **Alarm Nag** is being used then the accepted alarm will become active once more once the nag period has elapsed (and provided that the signal is still in an alarm state). This cycle will continue until the alarm condition is rectified.

Alarm Commenting

If **Alarm Commenting** is implemented then at the time the alarm is accepted a dialog will be displayed requesting a comment for the alarm that has just been accepted. This comment dialog does not interfere with the alarm display, and further alarms can be accepted before you begin entering comments.



The alarm time, signal name, description and alarm text is displayed at the top of the dialog. Enter the comment you wish to make in the box below. The text can be any length and can contain carriage returns.

If **Deferred Commenting** is allowed the dialog will have a second button labelled **Defer** which means that you will come back and enter a comment later, from the **Alarm Viewer** program. If deferred commenting is not allowed then the dialog cannot be exited without leaving a comment of some sort.

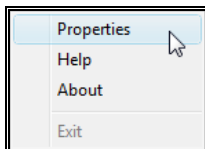
Security Considerations.

If security has been implemented then only an operator with the relevant privilege can accept alarms. When security is in operation the alarm banner message for an accepted alarm will additionally show the text “(accepted by <username>)” and the ID of the operator accepting the alarm will be written to the alarm and/or batch alarm logs. The alarm banner can be seen below when it was accepted by a default user.

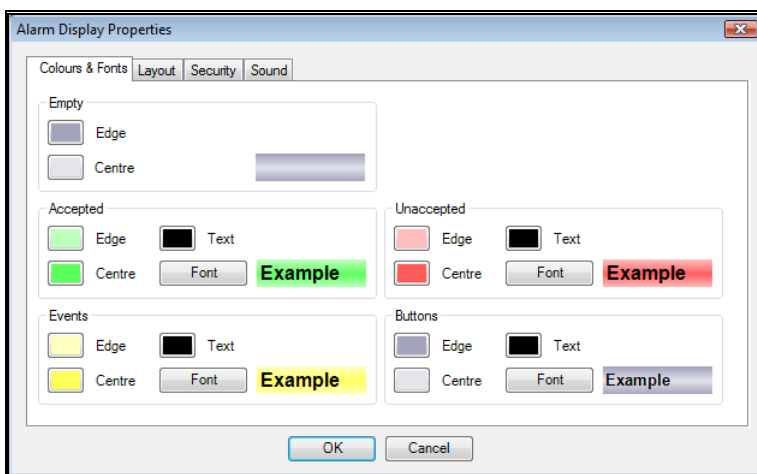


Alarm Display Properties

Select **Properties** from the context menu:

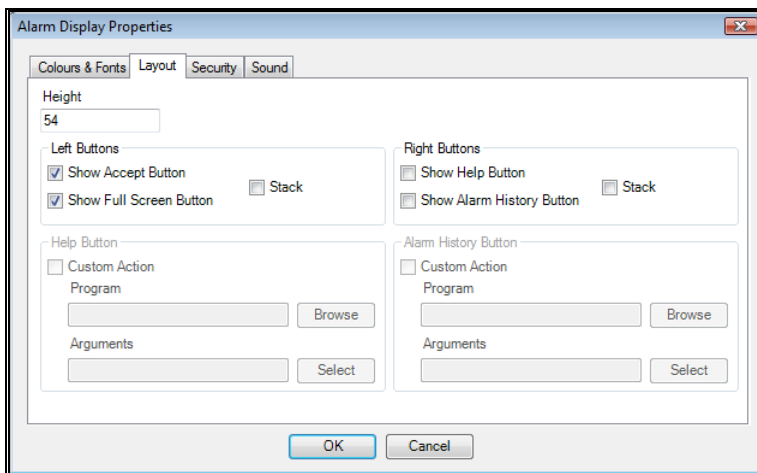


This brings up a dialog as shown below; there are four tabs to the dialog, they are Colours & Fonts, Layout, Security and Sound.

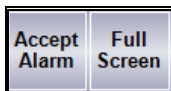


This allows you to define various attributes of the Alarm Display.

- **Empty.**
Select the background colour of the alarm banner when there are no alarms present, an edge and centre colour can be selected, if no shading is required simply select the same colour for the edge and centre. The effect of changes can be seen in the preview area.
- **Accepted.**
The background colour for accepted alarms can be selected and observed in the same way as for the Empty section. In addition the text colour and the font style can be selected for accepted alarms The effect of changes can be seen in the preview area.
- **Events.**
The background colour for Events can be selected and observed in the same way as for the Empty section. In addition the text colour and the font style can be selected for Events The effect of changes can be seen in the preview area.
- **Unaccepted.**
The background colour for unaccepted alarms can be selected and observed in the same way as for the Empty section. In addition the text colour and the font style can be selected for unaccepted alarms. The effect of changes can be seen in the preview area.
- **Buttons.**
The background colour for the buttons on the alarm banner can be selected and observed in the same way as for the Empty section. In addition the text colour and the font style can be selected for the buttons. The effect of changes can be seen in the preview area.



- **Height.**
The value entered here is the height on the screen in pixels that the alarm banner will occupy.
- **Left Buttons.**
As default the **Show Accept Button** and **Show Full Screen Button** are ticked, this places two buttons to the left of the alarm banner that can be used to accept alarms (as an alternative to double clicking). The two buttons can be seen below:



If the **Stack** button is checked then the two buttons above will be stacked so that they fit into half the space. This can be seen below; it is worth noting that the font size on the Buttons section of the **Colours & Fonts** tab will need reducing in order for the text to fit into the boxes as can be seen below.



- **Right Buttons.**
As default there are no buttons to the right of the menu bar. Checking the **Show Help Button** and/or the **Show Alarm History Button** boxes will place up to two buttons on the right hand side of the

alarm banner. The Show Help button will load this section of the manual up and the Show Alarm History button will start the Alarm History Viewer program. Unlike the two left hand buttons these two can be configured to run other programs. If the stack box is checked then the buttons are stacked in a similar way to the left buttons; below are the buttons in their default arrangement on the left and stacked on the right.

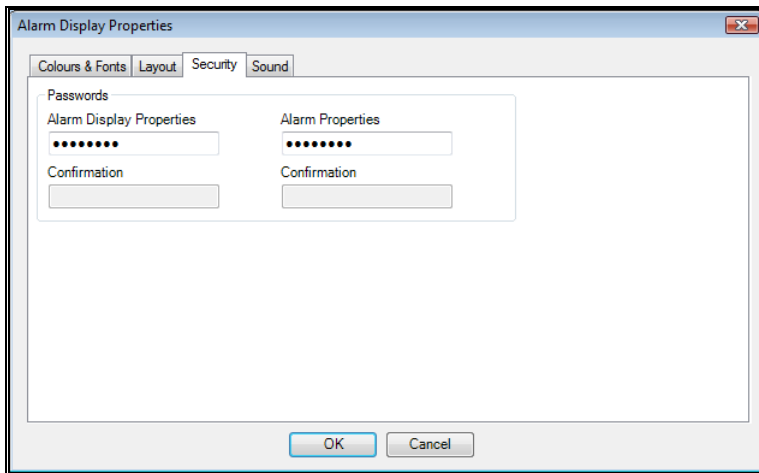


Normal

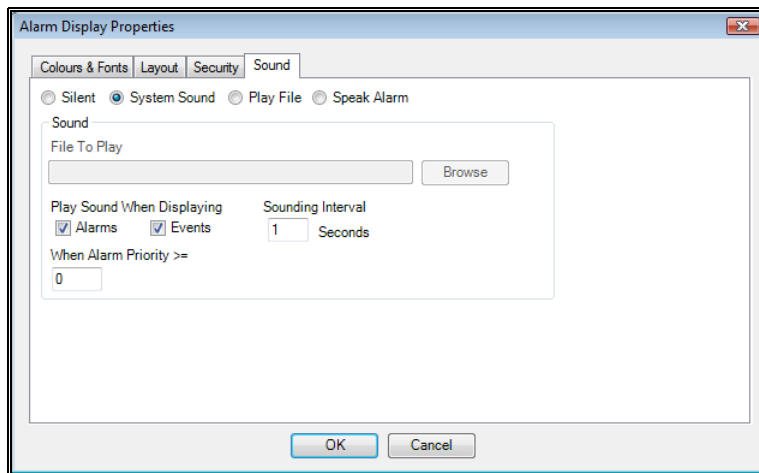


Stacked

- Help Button.**
If the Help Button is activated on the alarm banner then this section allows custom actions to be applied to the Help button. Other programs can be run in place of the prodigy help utility; to configure another program to run simply check the **Custom Actions** box and select the appropriate program. Any arguments that need to be sent to the programs can be entered; the arguments are the same as the parameters that are described for buttons in display builder.
- Alarm History Button.**
This is the same as for the **Help Button** only the section is only available when the Alarm History Viewer button is activated.



- Passwords.**
This allows you to password protect the **Alarm Display Properties** dialog and the **Alarm Properties** of signals if security using privileges is not set up on a system. To implement simply enter the required password into one of the boxes and as you do so the **Confirmation** box will also become available. Enter the password again into the **Confirmation** box. The system will check that both passwords match when **OK** is pressed.



- **Sound.**
Indicates how the alarm will be notified aurally.
- **Silent.**
There will be no audible warning.
- **System Sound.**
The standard PC “beep” will be played on the PC’s speaker.
- **Play file.**
A sound file of the standard .wav or .mp3 format will be played. This necessitates having a sound card and speakers in the PC. The Browse button allows you to select the relevant file.
- **Speak Alarm.**
The text that is written into the alarm banner will be converted into a sound file and “spoken” by the PC. This necessitates having a sound card and speakers in the PC.
- **Play Sound When Displaying.**
Indicates what will cause a sound to be played.
 - **Events.**
If this button is clicked the sound will be played when events occur, otherwise it will not.
 - **Alarms.**
If this button is clicked the sound will be played when alarms occur that are of greater or equal priority to the value entered in the **When Alarms Priority** >= box. In order to have the sound played for all alarms enter a priority of 0.
 - **Sounding Interval.**
This is the time between sound files been played and then repeated if a signal is still in alarm; if Play File or Speak Alarm are selected then this time needs to be greater than the longest possible time that an alarm is likely to sound.

Full Screen Mode

Pressing the **Full Screen** button on the Alarm Banner brings up a full screen dialog that contains the current state of all signals that are in an alarm condition. This window may contain a vertical scroll bar if there are too many signals to fit on one screen. Closing the window or pressing the **Full Screen** button again will remove the full screen window and return to the two line banner display.

The full-page alarm display is as shown below with most columns selected; some are off the right of the screen and would require the use of the scroll bar to be seen.

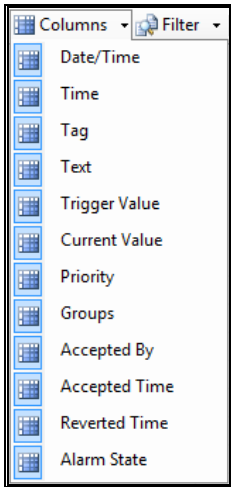
Date/Time	Tag	Text	Pr...	Trigge...	Curren...	Groups	Accept...	Accepted ...	Rev...
09-07-2008 08:35:32	LVLABLO	Acid Bath Low Probe was On	0		0	Status, Acid Bath			08:35
09-07-2008 08:35:57	AA	Waiting for operator to enter production details	0		<Unk>	None			
09-07-2008 08:37:18	C1MV	Zone 1 Measured Value is High	0	664.38	762.86	Fusion React, Temp Control			
09-07-2008 08:37:19	C2MV	Zone 2 Measured Value is High	0	663.52	762.86	Fusion React, Temp Control			
09-07-2008 08:37:20	C2MV	Zone 2 Measured Value was Very High	0	794.43	762.86	Fusion React, Temp Control			08:37
09-07-2008 08:37:26	C3MV	Zone 3 Measured Value is High	0	650.96	762.86	Fusion React, Temp Control	default user	09:33:19	

The full list of columns are as follows:

- **Date/Time.**
The date and time at which the alarm condition occurred.
- **Time.**
The time at which the alarm condition occurred.
- **Tag.**
The tag name of the signal that is in alarm.
- **Text.**
The alarm text.
- **Trigger Value.**
The value of the signal when it triggered the alarm.
- **Current Value.**
The value of the signal now..
- **Priority.**
The alarm priority.
- **Groups.**
Which groups the signal belongs to.
- **Accepted By.**
If users are implemented this displays the user who accepted the alarm.
- **Accepted Time.**
The time when the alarm was accepted.
- **Reverted Time.**
The time when the signal stopped been in alarm.
- **Alarm State.**
The state that the alarm triggered, this could be very low, low, high or very high.

Selecting Columns For The Full Screen Display

There is a Columns drop down list on the toolbar of the full screen alarm banner (see below). Each of the items on the list represents the columns that can be seen in the full screen view; to add or remove a column click on the column name that you want to add or remove.

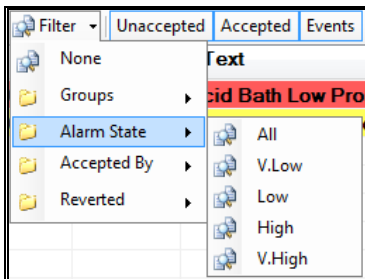


Only the **Date/Time** OR the **Time** column can be displayed at any one time; they both cannot be displayed at the same time.

You can click on any column header to sort the display by that column. Clicking the column again reverses the sort order.

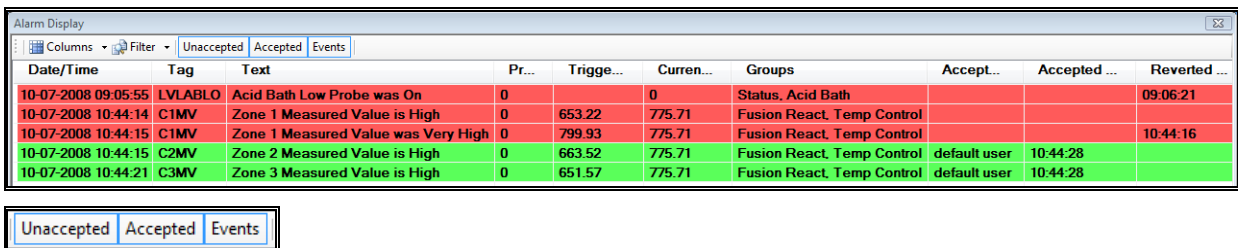
Applying Filters To The Full Screen Display

There is a Filter drop down list on the toolbar of the full screen alarm banner (see below). Filters can be applied to the current alarms to make viewing the most critical alarms easier.



Filters can be applied to alarms based on the groups signals belong to, the alarm state, who accepted the alarm or if an alarm has reverted or not. To apply a filter click on the appropriate filter section and select the filter that you want to apply; multiple filters can be applied by simply clicking on multiple filter criteria. The removal of a filter can be done either by clicking on the filter that needs to be removed, or if all filters are to be removed click on **None** that can be seen in the above image.

Further filtering can be applied that allows you to select what combination of unaccepted alarms, accepted alarms and events will be displayed. Three text buttons are available on the toolbar to allow selection of the appropriate items.



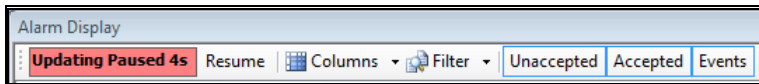
Clicking on the **Accepted** button will cause the highlighting around the Accepted text it to be removed and all accepted alarms will no longer be displayed on the full screen display. An image of how the full screen display looks when in this state can be seen below. To restore the full screen display so it shows the accepted alarms click on the accepted button, it will become highlighted and any accepted alarms that are still activate will be displayed again. Alternatively the full screen window can be closed and re-opened to restore the Accepted alarms.

Date/Time	Tag	Text	Pr...	Trigge...	Curren...	Groups	Accept...	Accepted ...	Reverted ...
10-07-2008 09:05:55	LVLABLO	Acid Bath Low Probe was On	0		0	Status, Acid Bath			09:06:21
10-07-2008 10:44:14	C1MV	Zone 1 Measured Value is High	0	653.22	775.71	Fusion React, Temp Control			
10-07-2008 10:44:15	C1MV	Zone 1 Measured Value was Very High	0	799.93	775.71	Fusion React, Temp Control			10:44:16

Selecting Alarms

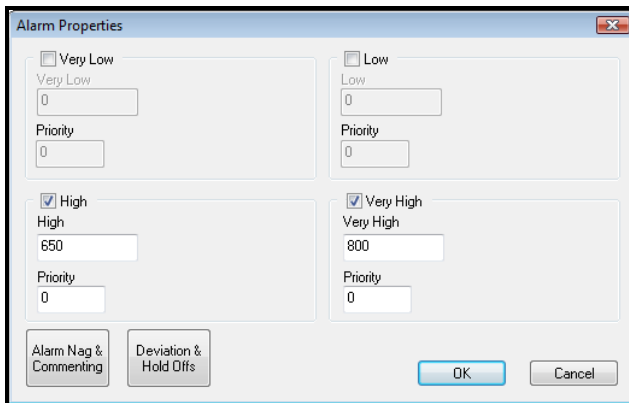
You can select any number of signal alarms by clicking on the first one and then either shift-clicking on a second one to select a range, or by ctrl-clicking on subsequent signal alarms to select them individually. You can then right-click over a signal and select “**Accept**” from the popup. All selected signals will be accepted.

When a signal is selected the toolbar has a countdown timer placed on the left hand side; this indicates the time period over which updates of the current signal values will not occur. During this time you can perform multiple selections of alarms, when the countdown timer reaches zero the selections that have been made will automatically be deselected and the signal values will begin to update again.



Alternatively clicking the resume button that appears when multiple alarms are selected will start the system updating the current signal values again.

If you select only one signal, you can right-click on it and select “**Properties**” from the menu. This will allow you to change the alarm properties for that signal, as shown below.



Any changes made are saved to disk and will be the new alarm limits for that signal even if Prodigy is restarted.

Events

Event messages are generated by various pieces of software and are an indication that something has happened or that some action needs to be performed (e.g. *Waiting for vessel to fill*). They appear in the alarm display and remain there until they are accepted or (in the case of timed events) they remove themselves.

Alarm Recording.

Alarm recording can be enabled on a per-signal basis via the **Signal Database Configuration** dialogs. It is achieved by selecting the relevant options in an alarm strategy and then assigning this strategy to a signal

where the recording of alarms is required. Alarms may be recorded to a stand-alone alarm log and to a batch database, where they are recorded as part of the currently running batch. It is possible to have alarms recorded to one or both of these locations.

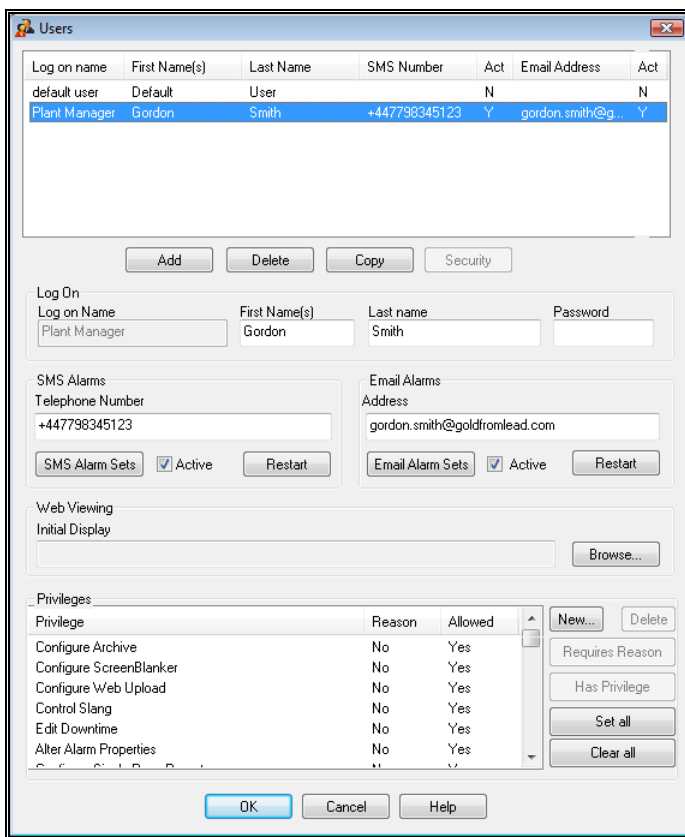
When an alarm condition arises the signal name, time, alarm state, signal value and the text string describing the alarm condition (as displayed on the alarm banner) are written to the alarm database. When the alarm condition is acknowledged this recording is repeated along with additional information stating that the record indicates an acceptance (and optionally the ID of the accepting operator, if security is implemented).

Events are logged to the alarm databases in the same way, with a flag indicating that they are events and not alarms.

SMS & Email Messaging

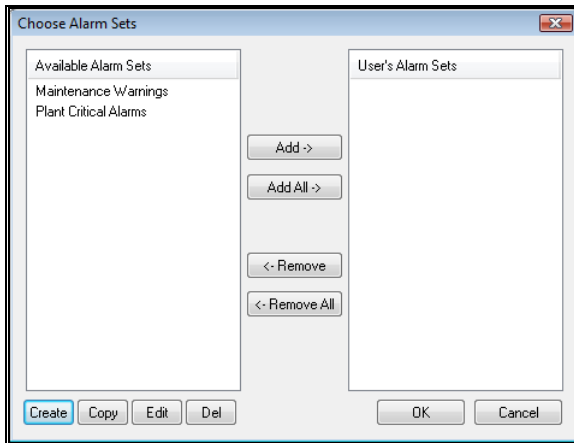
When an alarm condition occurs you may need to notify some member of staff who is either off site (for instance an on-call engineer) or who may be elsewhere on the site. Prodigy can do this automatically by sending an SMS message to a mobile phone that is capable of text messaging. Alternatively the alarm message can be sent via Email to someone either on-site or off.

To configure the SMS/Email facility, use the *User Configuration* program.



Click on the user for whom you want to configure SMS/Email messaging. To configure SMS alarms enter their mobile phone number (using the international format, e.g. +441234123456 and **NOT** 01234123456) into the **SMS Alarms Telephone Number** box and then click the **SMS Alarm Sets** button. For Email enter their email address into the **Email Alarms Address** box and click the **Email Alarm Sets** button.

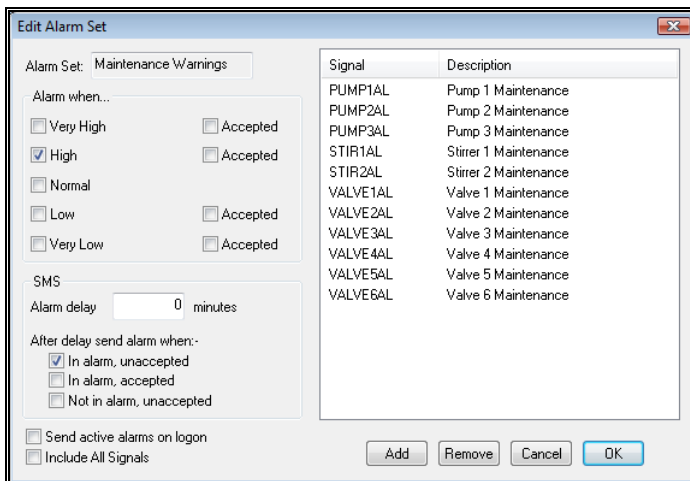
In both cases the *Choose Alarm Sets* dialog is displayed (see below).



An Alarm Set is a list of alarms. These may be grouped according to the area of plant, by how critical they are, by the engineer or engineers responsible for certain items of plant, and so on. Double click on an alarm set in the left-hand list to transfer it to the right-hand list and vice versa. All Alarm Sets in the right-hand list are assigned to the current user.

You can create a new Alarm Set by clicking the *Create* button. This will prompt you for the name of the new Alarm Set.

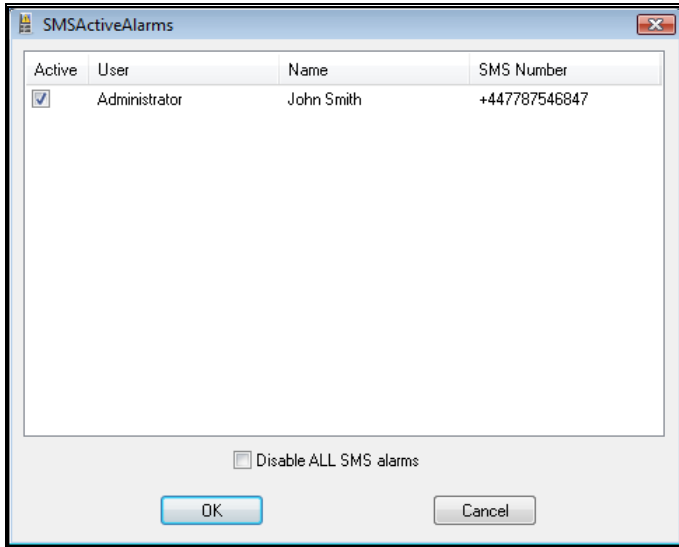
After entering the name the *Edit Alarm Set* dialog will appear (below). This same dialog is used if the *Edit* button is clicked from the *Choose Alarm Sets* dialog (above).



- Add signals to the Alarm Set by clicking the **Add** button and choosing the required signal from the SigScan dialog.
- Remove signals from the list by selecting the required signal in the list and then clicking the **Remove** button..
- Select the alarm states that you want to be notified about (Very High, High, Accepted, etc). An SMS message will be generated if **any** of the signals in this Alarm Set enters one of the conditions that have been checked.
- If you want the alarm set to include all signals, simply check the **Include All Signals** box.
- If you want a delay to occur before an alarm is sent via SMS then enter the time in minutes in the alarm delay box. If a value is entered for the delay then three states of alarms can be sent to your mobile device they are; In Alarm (Unaccepted), In Alarm (Accepted) and Not In Alarm (Unaccepted).
- If you want to be notified of all current alarms when you log-in (via your phone or Email, see later) then check the **Send active alarms on logon** box.

Once you have finished click **OK** to return to the **Choose Alarm Sets** dialog. Once you have selected all relevant Alarm Sets for the current user click OK to return to the Users Configuration dialog.

SMS Active Alarms



On the Users Configuration dialog next to the buttons for **SMS Alarm Sets** and **Email Alarm Sets** are check boxes labelled **Active**. Checking or clearing these boxes makes a user either available or unavailable for the relevant facility. There is also a global facility **SMS Active Alarms** which is accessible from Quick Start or from the Menu Bar (if so configured). This provides a facility to allow users to be activated or deactivated from SMS Alarms without providing access to the full Users Configuration program. Note that this program also contains a flag to globally disable **ALL** users from SMS Alarms, so you **must ensure** that this flag is cleared when setting up an SMS Alarm system.

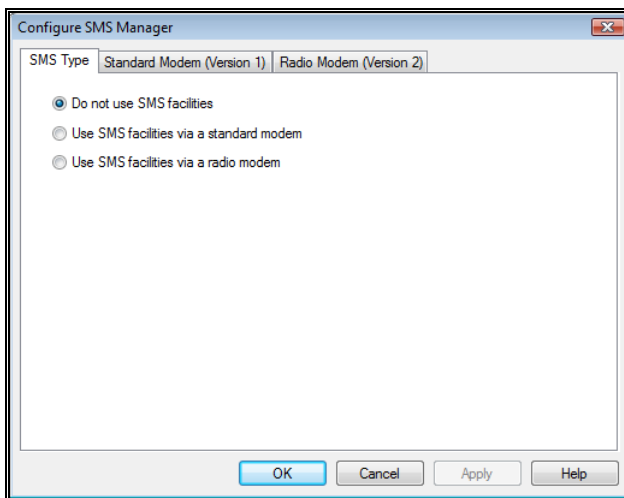
When an alarm condition occurs in a signal, **Prodigy** looks to see which alarm sets (if any) the signal belongs to. For each alarm set that it finds it will look to see which users have this alarm set selected. It will then attempt to send an SMS message to those users giving the following information:

- Machine name on which the **Prodigy** system that generated the alarm is running.
- The date and time of the alarm.
- The text of the alarm message.
- An alarm identifier (a unique code identifying this alarm, that may be used to **Accept** the alarm).

If this user is going to use the interactive facilities allowed with GSM modems, then they must be given the appropriate privileges.

Configuring SMS Services

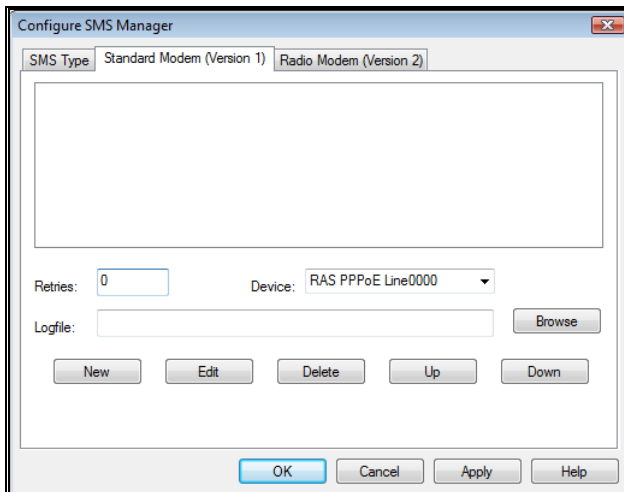
Prodigy sends SMS messages by dialling out to an SMS service provider, either via a standard modem or via a GSM modem. **Select SMS Manager Configuration** to display the dialog shown below.



Selecting The Service Type

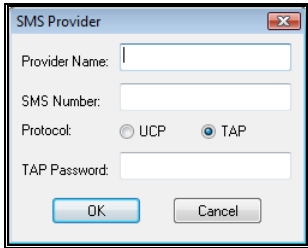
Select the service type you wish to use. The full interactive functionality of SMS services is only available with a GSM radio modem, and this also gives the most reliable method of operation. A standard modem allows only the more limited functionality of receiving alarm notifications via an SMS message and offers no interactive facilities.

Configuring A Standard Modem

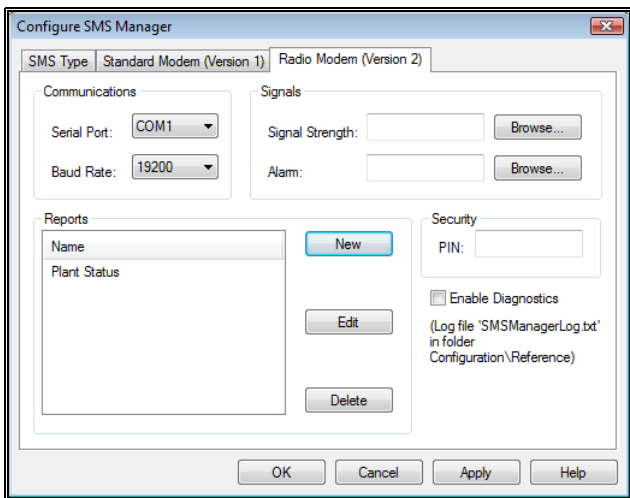


When using a standard modem you must use a *service provider* to which the messages are transmitted via the landline modem. The service provider then transmits them to their final destination as a proper SMS message. Prodigy comes configured with a set of service providers but you may need to amend this list (details given below). You can specify the number of times the system will attempt to send the message if it encounters an error, along with the device through which the message is to be sent. Additionally you can specify a log file into which information will be written as the system attempts to send SMS message.

Click **New** to create a new **provider** entry, or **Edit** to modify an existing entry. In both cases the dialog shown below is displayed. Enter the relevant information into the fields (you may need to contact your system administrator or your Prodigy supplier for assistance).



Configuring The Radio Modem

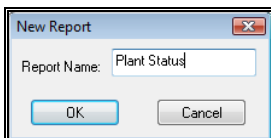


In the **Communications** section specify the COM port to which the radio modem is attached and the baud rate at which it is communicating.

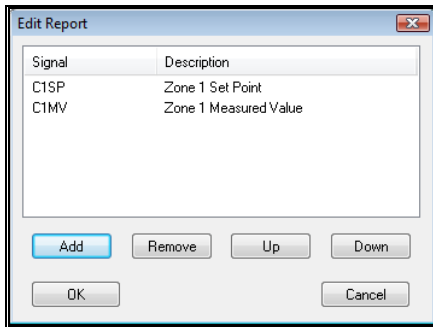
In the **Signals** section (optionally) specify a signal that will be used to display the signal strength as perceived by the radio modem, and a signal that will be alarmed if the computer loses communications with the radio modem. (The signal strength is reported in the range -113dBm to -51dBm , where -51 is better (stronger) than -113).

In the **Reports** section you can create a list of reports that are available to the SMS user. Each report consists of any number of signals, and when an SMS user calls up this report they are sent the name and value of each signal that is in the report. If the resulting output of this report is longer than is allowed in an SMS message, it is split into multiple messages. This facility allows operators to quickly inquire about the state of groups of signals without having to inquire about them individually.

To create a Report click **New** which will display the following dialog



To edit an existing Report click **Edit** – in both cases the dialog below is displayed.



Add or remove signals as required, and arrange them in the desired order by using the buttons on this dialog. Click **OK** when the list is complete.

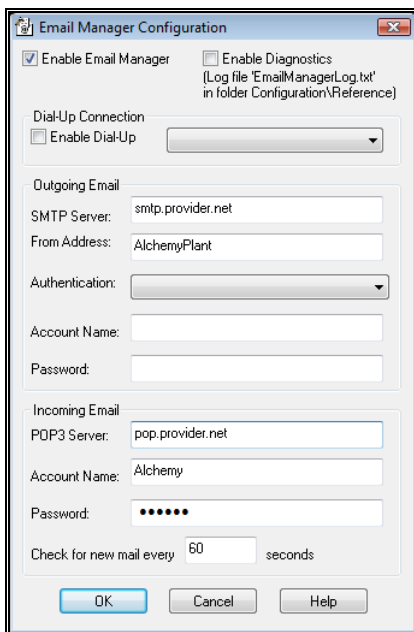
Interactive Facilities With The Radio Modem

When using a GSM radio modem there are a variety of interactive facilities allowed. The operator can

- Receive alarm notifications.
To do this by assigning the GSM phone with a list of alarm sets (described earlier in this manual section). Each alarm is assigned a unique ID that must be used when accepting that alarm.
- Accept alarms (requires the appropriate privilege).
To do this the operator must send the SMS message **A alarm_id** (the alarm_id is given with the alarm notification).
- Inquire about the value of a signal (or signals via a **report**) (requires the appropriate privilege)
To do this send the SMS message **D signal_name [signal_name...]** (signal_name is exactly as specified in the signal database, and you can enter any number of signals separated by a space).
To ask for a report send **G report_name**.
- Set the value of a signal (requires the appropriate privilege).
To do this send the SMS message **W signal_name value password**. Note that with this facility your password (as specified in the Users configuration in Prodigy) is required as an extra security measure.
- Send a text message to the Prodigy system.
To do this send the SMS message **T message**. The message will be displayed in a dialog box on the Prodigy display. If the sender is known (the phone number is in the SMS telephone Number section of the User Configuration) then their name is show in the dialog, otherwise the phone number of the sender is shown.

Configuring Email Services

Prodigy sends Email messages by communicating with an Email server. Configure the Email server's settings by running **Email Manager Configuration** from the Menu Bar or QuickStart.



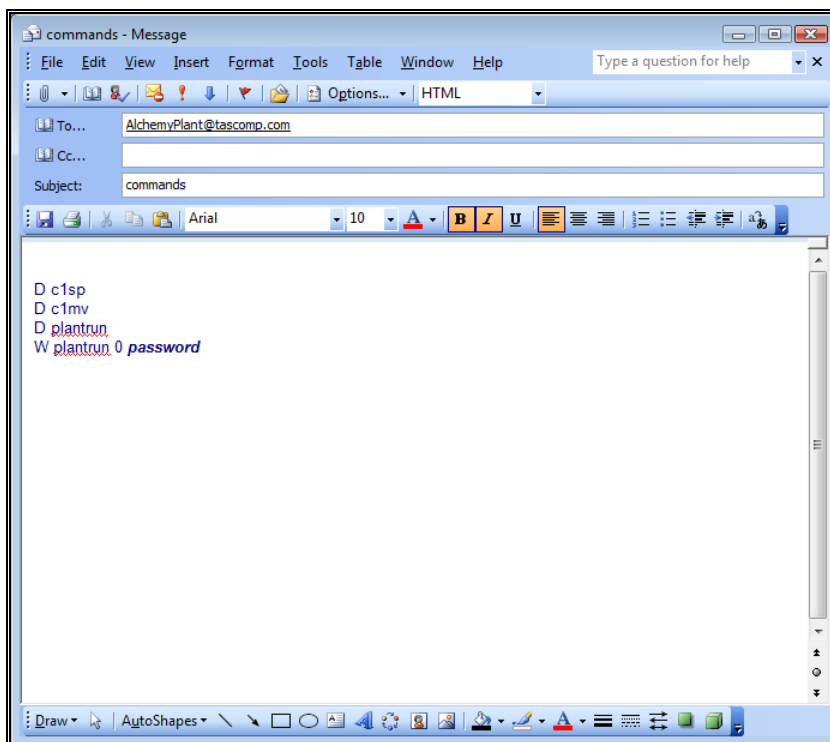
- **Enable Email Manager.**
This check box is used to globally enable or disable email alarms.
- **Enable Diagnostics.**
If checked the Email manager creates a log file that records any Emails that are sent by the system or received from users.
- **Dial-Up Connection.**
If you connect to the Internet via a dial-up connection (modem), check this box and select the dial-up connection to be used.
Note due to circumstances that are beyond our control, a dial-up connection cannot be used if Prodigy is running as a service.
- **SMTP Server.**
The name of your SMTP server at your ISP (e.g. smtp.provider.com).
- **From Address.**
The name you want the Emails to be from (e.g. AlchemyPlant) when Prodigy sends them out and also the name to which Emails should be sent in order for them to be processed by the Prodigy Email client. In this example, if our domain is tascomp.com then mails will be from AlchemyPlant@tascomp.com.
- **Authentication.**
The level of authentication that the SMTP server requires.
- **Account Name/Password.**
The account name and password required to log in to the SMTP server (if required).
- **POP3 Server.**
The address of the POP3 server (e.g. pop.provider.com).
- **Account Name/Password.**
The account name and password used to log in to the POP3 server.
- **Check for new mail every:**
How often to check for incoming mail (does not affect outgoing mail, which is sent as soon as it is generated).

Prodigy acts as an Email client, much as Microsoft Outlook/Express™ does. If you already use an Email client to access your provider (such as Outlook/Express) then these settings will already be configured for that client so you can find out the correct settings by looking in there.

Interactive Facilities With Email

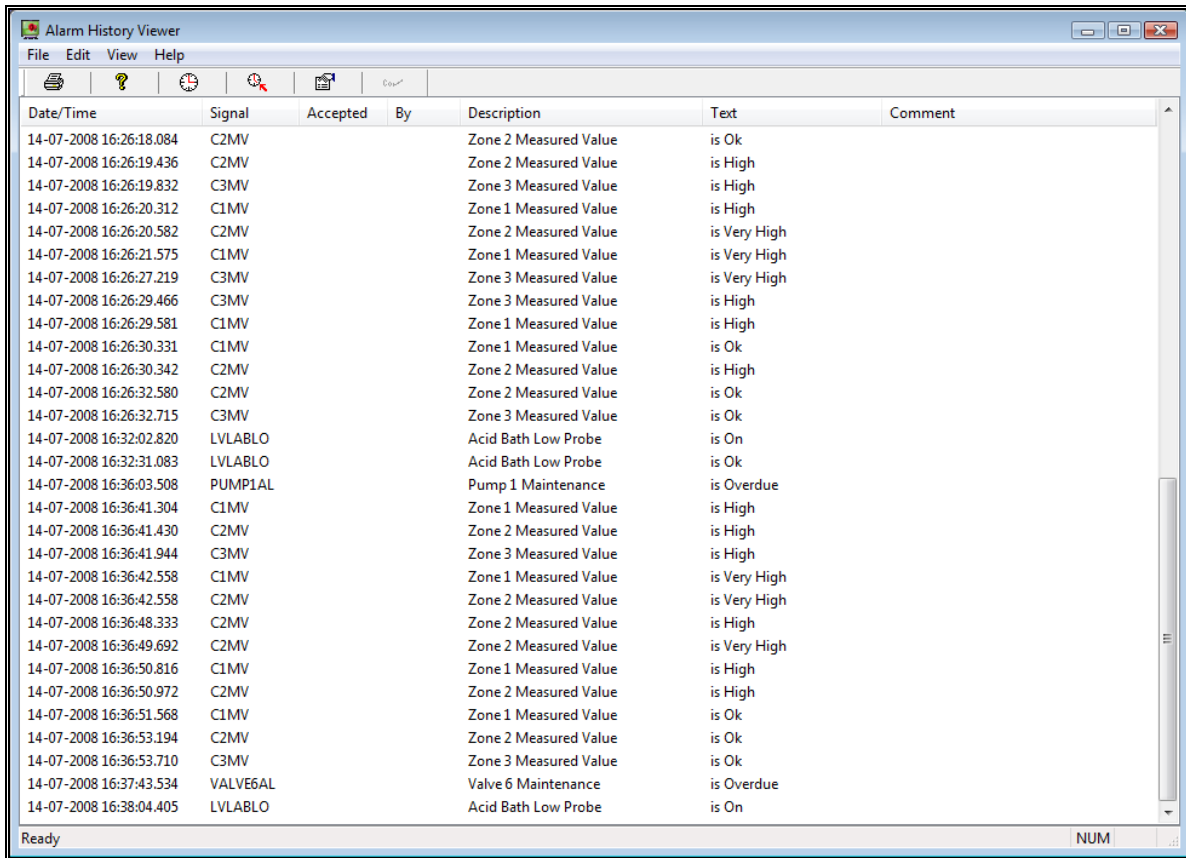
When using Email there are a variety of interactive facilities allowed. The operator can

- **Receive alarm notifications.**
Do this by assigning the Email with a list of alarm sets (described earlier in this manual section). Each alarm is assigned a unique ID that must be used when accepting that alarm.
- **Accept alarms** (requires the appropriate privilege).
To do this the operator must simply reply to the Email message. No further information or instructions are required in the mail.
- **Commands.**
A variety of commands can be sent via Email. To do this simply send an Email with the subject **commands** and include any number of the following commands (one per line) in the body of the mail.
 - **D <signal name>.**
Returns the signal value.
 - **W <signal name> <value> <password>.**
Sets the value of the signal.
 - **I <password>.**
Log in to receive alarms.
 - **O <password>.**
Log out to stop receiving alarms.
 - **H.**
Request a mail containing help (shows all of the above commands).

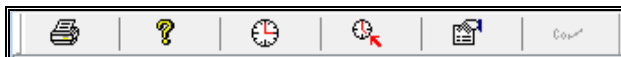


Alarm History Viewer

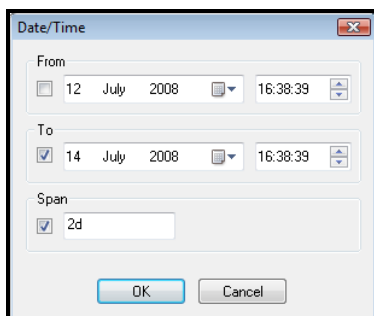
The Alarm History Viewer is an interactive tool for viewing the alarm history database.



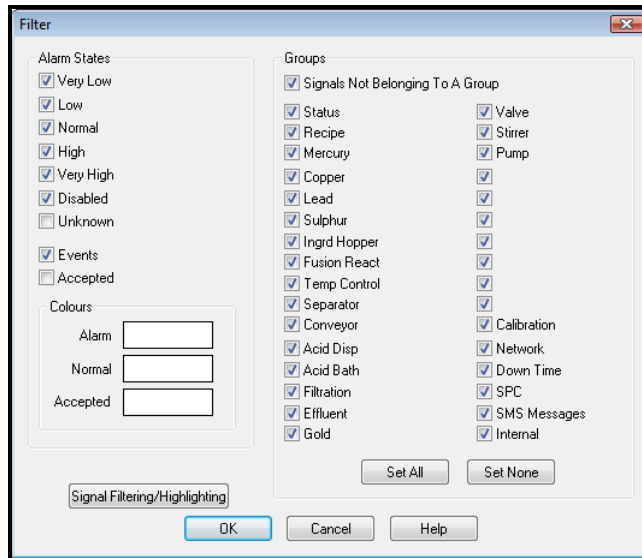
The toolbar at the top of the window controls the functionality.



- **Print.**
Prints the currently selected records, taking into account all filtering that is currently in place. The printout is in a similar collimated form to that shown on the screen, and all selected records are shown, regardless of if there are too many to see on the screen at once (printing can occur across multiple sheets).
- **Help About.**
Displays the Help About box.
- **Select Time.**
This button allows you to select the range over which you want to view alarms. Select any two of **From**, **To** and **Span** to specify the range.



- **Refresh.**
This causes the event viewer to re-read the log. New alarms may have occurred since the viewer was opened, and this will cause them to be displayed (provided they occurred within the timeframe currently being viewed of course).
- **Filter.**
This displays the filter dialog (shown below).

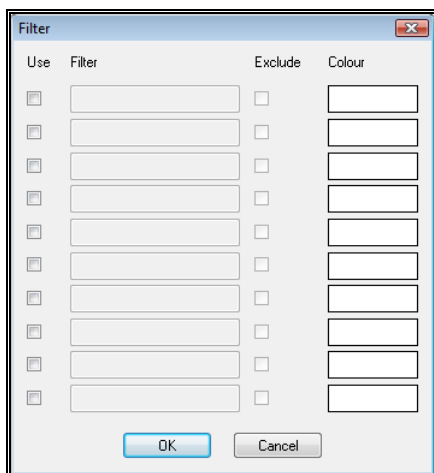


Filtering is quite simple:

- Select the **Alarm States** that you want to see.
- Select if you want to see **Disabled** and/or **Unknown** states too.
- Select if you want to see **Events** and alarm **Acknowledged** records.
- Select the colours you want to use for records that show **Alarm**, **Normal** and **Accepted** states.
- Select all of the groups that you are interested in (and check **Signals Not Belonging To A Group** if you also want to see signals that do not belong to any group).

Signal Filtering/Highlighting

This button display the following dialog:



- Check the **Use** box to enable or disable a filter definition.
- Enter the **Filter**, which is simply the name of a signal to be filtered.
- Check or clear the **Exclude** box.
If it is checked the signal name entered under **Filter** will not be shown in the Alarm History Viewer.

If it is not checked then this signal **will** be shown, but only this signal and all other signals that have filters with no **exclude**.

- Select a **Colour** for the signal (if it is not being excluded). When this signal is displayed on the Alarm History Viewer its row will be highlighted in this colour.
- If a **Use** box is checked and no **filter** is given and **exclude** is not checked (as in the final entry, below) this causes **all** records that do not have a specific exclusion to be displayed. (If this final entry were not set up, our display would show only entries for the **BATCHID** signal. As it is, it shows **all** records, but with **BATCHID** highlighted).
- **Sorting.**
To sort the data in different ways simply click on the relevant column heading. To reverse the sort order, click the column again. Note that the data can only be sorted by **Date/Time** or **Signal**. If the data is sorted by Signal the secondary sort is always on Date/Time.

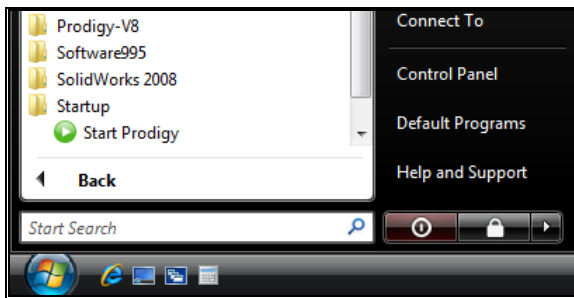
Note that all filters etc that are applied will remain in effect the next time the Alarm History Viewer is run, so on subsequent runs make sure that the filtering is still as you expect it to be.

Display Launcher

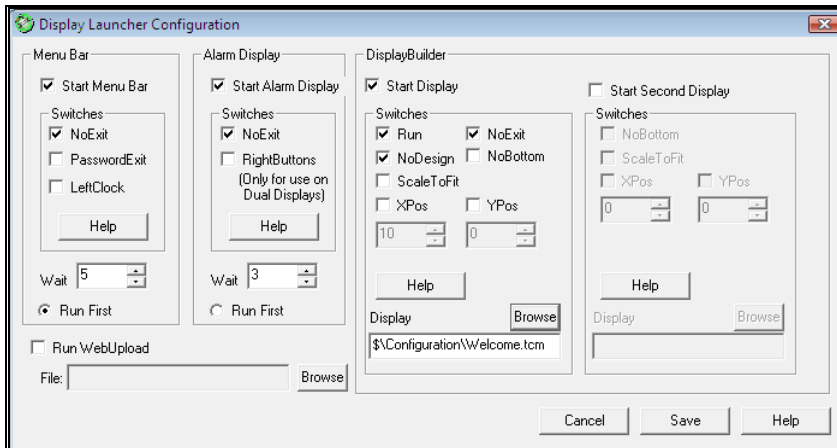
Display Launcher

The Display Launcher (Start) starts up Prodigy Services if they are not already running and then activates the common visual components of Prodigy – the Menu Bar, the Alarm Display and a **Display Builder** display (two displays if using a dual screen system). Which components are launched and which Displays are displayed is defined by Display Launcher Configuration.

Display Launcher (Start) can be added into the Startup folder of the Windows Start menu so that it is fired off automatically when the PC is logged in. Alternatively it can be started from the Prodigy Control Panel, from Quick Start or from the Prodigy-V8 group of the Windows Start Button.



Display Launcher Configuration



This is the Display Launcher Configuration dialog. The options are as follows:

- **Start Menu Bar.**

Check this box if you want to launch the Menu Bar.

NoExit.

Prevents the user from being able to exit the Menu Bar (the Exit option will not be available on the right-click menu).

PasswordExit.

Allows the user to exit the menu bar if they give the correct password.

LeftClock.

The clock (and logon button) is placed at the left of the Menu Bar instead of the right.

Wait.

Enter a number of seconds for the system to wait before it launches the next component (Alarm Display or Display Builder Display).

This is necessary to avoid problems caused by the Menu Bar trying to reserve screen space at the same time as another application.

Run First.

Will cause the menu bar to be run first and appear below the alarm banner.

- **Start Alarm Display.**

Check this box if you want to launch the Alarm Display.

NoExit.

The user will not be able to exit the Alarm Display (the Exit option will not be available on the right-click menu).

RightButtons.

This puts the **Accept Alarm** and **Full Screen** buttons at the right hand side of the screen rather than the left. It is intended mainly for Dual Display situations.

Wait.

Enter a number of seconds for the system to wait before it launches the next component (Display Builder Display).

This is necessary to avoid problems caused by the Menu Bar trying to reserve screen space at the same time as another application.

Run First.

Will cause the alarm banner to be run first and appear below the menu bar.

- **Start Display.**

Check this box if you want to launch a display.

- **Run.**

The display will be launched in Run mode (rather than Edit mode). This is the most sensible option when starting Display Builder from Display Launcher.

- **NoExit.**

The user will not be able to exit from Display Builder. Must be used in conjunction with Run or NoDesign.

- **NoDesign.**

The user will not be able to access the Design mode of Display Builder.

- **NoBottom.**

By default Display Builder Displays go to the bottom of the screen's Z-order, meaning that all other windows appear over the Display Builder Display. This is the best course of action in single-screen displays as it prevents other windows from being obscured by the Display Builder display, which occupies the full screen. (Exceptions to this are when Display Builder is in edit mode and when displaying a Historical Trend).

If you have a dual-display system and have a Display Builder Display on each of the monitors then you will have to select NoBottom for one of them, otherwise they constantly flicker as each one constantly tries to place itself at the bottom of the Z-order.

- **Scale to Fit.**

Scales the display up or down so that when it is launched so it fits the screen.

- **XPos, YPos.**

When using a dual display you can use the XPos and YPos fields to specify where the Display is placed. Specify an XPos value that is sufficient to have the Display appear on the right-hand or left-hand monitor, as appropriate. YPos should be sufficiently far from the top of the screen to not obscure the Alarm Display and/or Menu Bar.

- **Display.**

Browse for the Display that is to be launched.

- **Start Second Display.**

The options are as for **Start Display**.

- **Run Web Upload.**

Here you must enable Web Uploads and specify the **Web Upload Configuration** file, this will cause the Web Upload program to run and upload files from the appropriate directory as they appear. For more information consult the **Web Upload** section of this manual.

A Word About Services

Prodigy consists of a suite of programs that interact with each other to provide the necessary functionality of a modern SCADA system. The user sees some of these programs (e.g. Display Builder, Signal Database Configuration) but others are background processes that have no user interface. These background processes form the core of Prodigy, handling such things as plant I/O, signal management, alarm management, batch recording, data logging and so on.

Before any of the visual aspects of Prodigy can be used all of the core processes must be running. Indeed, it is possible that no visual aspects of Prodigy may be required, the system simply providing monitoring facilities through data logging and BDR, in which case only the background processes are required.

It is essential that the background processes are running all of the time that the Prodigy system is in use, therefore when the computer is powered up the background processes should be started automatically. If any of the background processes should "die" for any reason, Prodigy may become inoperable and a restart may be required. The user must be informed if such a situation arises.

Different users may need to log in and out of the computer and in doing so may start and stop any foreground applications that are running. It is essential that this process does not interfere with the background processes, which should continue uninterrupted, regardless of who, if anyone is logged in.

To provide the facilities described above, Prodigy makes use of a Windows facility known as **Services**. Services provide a mechanism for starting processes that must be running whenever the computer is running but which are completely independent of whether or not any user has logged onto the computer. Operating systems like Windows NT™, 2000™, XP™ and Vista™ all run a variety of services that the computer user largely need to know nothing about

On Prodigy Workstation systems Prodigy must **not** be started as a service.

What Does Prodigy Service Start

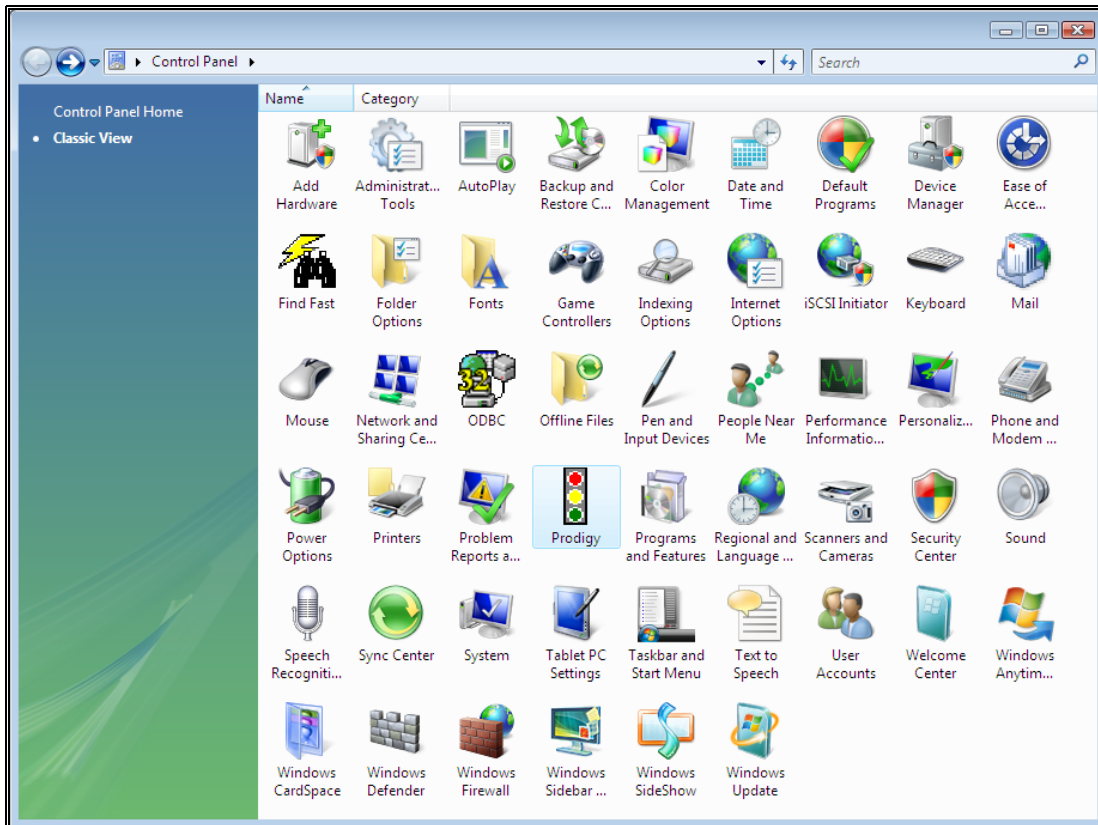
When the Prodigy Service is started:

- All of the core processes are started.
- The signal database is initialised.
- Any of the **Optional Services** that are selected are started.
- Any drivers that are required are started. (The Driver Configuration program is used to specify which drivers are required).
- A list of automatic XForms can be run to initialise batch variables and/or signal values from recipe data (see the **XForms** tab of the **Prodigy Services** dialog, below).
- Any additional processes that have been specified are started, so long as they have been checked as being "Active". (See the **Additional Programs** tab of the **Prodigy Services** dialog below).

Access to Prodigy Services

Prodigy Services are controlled through the Prodigy Control Panel, this can be accessed through the Quick Start application, the Windows Control Panel (see below) or the Prodigy Menu Bar if Prodigy is running and the operator has the correct privileges.

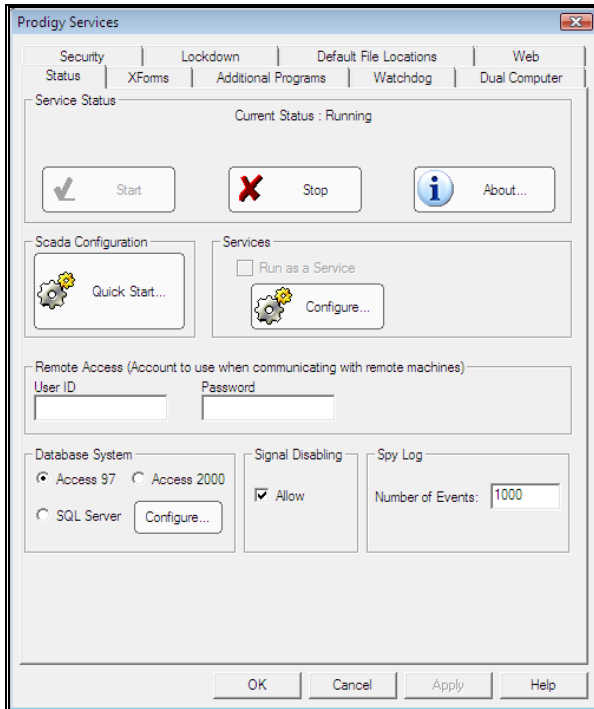
To gain access through the Windows Control Panel the user must double click the relevant icon. The screen shot below is for Windows Vista and The Prodigy Icon is highlighted.



Prodigy Control Panel

The Prodigy Control Panel provides a mechanism to stop and start Prodigy. In addition it is used to specify any additional programs, optional services, watchdog facilities, data archiving and backup criteria and dual computer support. This is all achieved via a tabbed dialog.

Service Status Tab



This tab has 7 groups of functions:

- **Service Status.**

There are two buttons to **Start** and **Stop** the services. Starting the services actually calls **Display Launcher** and so will also optionally start the **Alarm Display**, **Menu Bar** and **Display Builder**.

Above the buttons is an area for a message describing the current status of the services. This is usually either **Stopped** or **Running**, though during start-up it will change to show the current phase of the start-up procedure (**Starting Main Services**, **Starting Drivers** or **Starting Additional Programs**). If anything has gone wrong (i.e. part of the services has died) the message will show this too.

The **About** button shows the Prodigy About Box which gives information about the current version of Prodigy and the license being used.

- **SCADA Configuration.**

The **Quick Start** button launches the Quick Start toolbar, which has access to all of the user-accessible components of Prodigy.

- **Services.**

The **Run as a Service** check box allows you to specify whether Prodigy is run as a service or as an application.

The **Configure** button allows you to select which of the optional services will be run.

Note that not all of these options are available to all users – some are license dependent. The optional services are described later.

- **Remote Access.**

This is only required if Prodigy is started as a Service (Run as a Service box is checked) and your system will need to access files or directories on another. For instance:

SLANG or **BDR** are going to access files across a network.

The system is going to use **Historical Trending** to access archived data that is held on another computer (see **Data Location**).

The Automatic Archive facilities archive data to another computer.

The User ID and Password entered here must be valid on both this computer (the computer running Prodigy) and the computer on which the data is being held.

- **Database System.**

Prodigy can write Access™ or SQL database files in either Access 97, Access 200x or SQL Server format. Select the appropriate format for your organisation. For instance, if Office 2000 or above is used in your organisation select the Access 2000 option. If some people who need to access the data are still using Access 97 then select this option - you should always choose the lowest common denominator.

Once Access 2000 has been selected you cannot revert – any databases that have been created with the 2000 option active will not be able to be opened by 97. If you do need to revert to 97 contact Tascomp or your integrator for help and information.

SQL Server can be selected, this is not compatible with Access 97 or 200x databases however there is a tool that can convert between Access databases and the SQL format therefore if your company requires SQL this is an option. The configure button can be used to configure the SQL Server database.

- **Signal Disabling.**

When a signal is disabled it is treated as if it does not exist. Signals may be disabled when, for example, an area of plant is shut down for maintenance. The values coming back from that area may be meaningless and may trigger alarms that are not valid, or may trigger other parts of the system. By disabling the signals any knock-on effects are minimised and the value of the signal is recorded as simply being disabled.

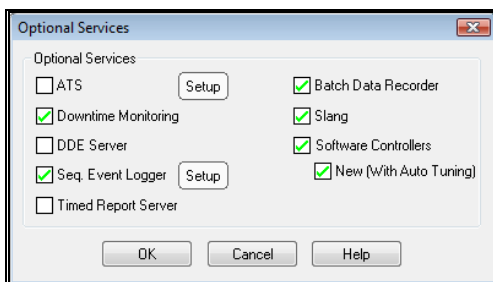
There is an overhead associated with disabled signals, especially at system startup, so if you are never going to use signal disabling you can uncheck the Signal Disabling box. You may also want to do this simply as a safety measure to prevent the accidental (or malicious) disabling of signals by operators.

- **Spy Log.**

Prodigy Spy logs informational and/or error messages from all of the components of a Prodigy system. This box controls how many log entries are maintained in the file. See later for details about using Spy.

Optional Services Configuration

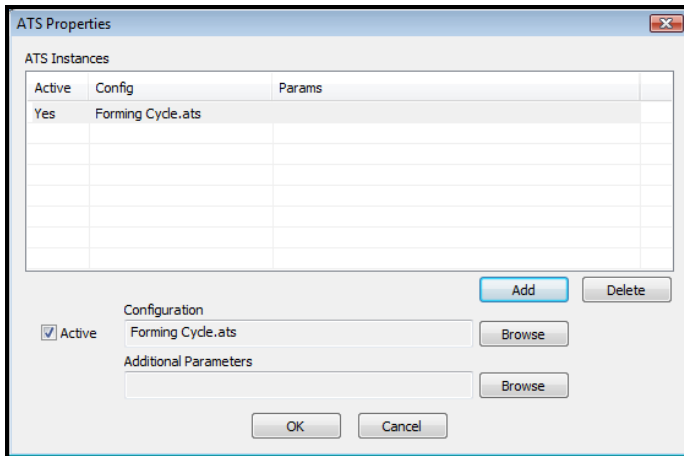
The optional service dialog is as below:



Check the box of those optional services that you want to run. *See the relevant section of the manual for more information on these services.* If Prodigy is already running, checking a box will start the relevant service immediately. Similarly, unchecking it will stop that component.

ATS

If **ATS** (Automated Test Sequence) is selected then it will need to be configured, which is done by pressing the **Setup** button. This displays the following dialog:

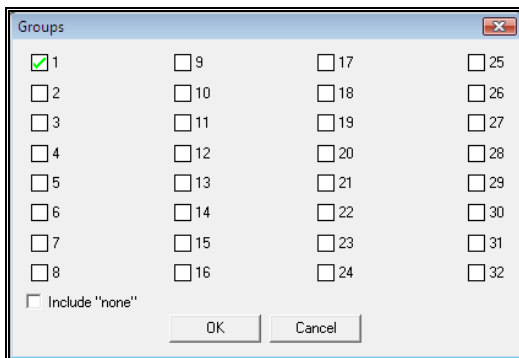


Add as many different configurations as you like to the ATS Instances list by using the **Add** button and filling in the relevant fields. Configurations may be omitted from running by clearing the **Active** flag if required. After making changes to this list exit the dialog by pressing **OK**. On returning to the Optional Services screen you must uncheck and then recheck the **ATS** flag in order to restart the **ATS** services. This is deliberately not done automatically in case a test sequence is already running. You should ensure that it is safe to do so before cycling the **ATS** flag.

*Note that **ATS** configurations must be created in advance by using the **Automatic Test Sequence Configuration** program.*

Sequence of Event Logger

If **Sequence of Event Logger** is selected then you will also have the option to configure the logger. Clicking the **Config** button will display the following dialog:



Check the groups that you want to log signal state changes for and also check the **Include “none”** box if you want to log state changes for signals that do not belong to any group. **Sequence of Event Logger** records changes of state of digital & finite state signals belonging to the specified groups. These can be viewed later by the dedicated viewing tool (**Event Viewer**).

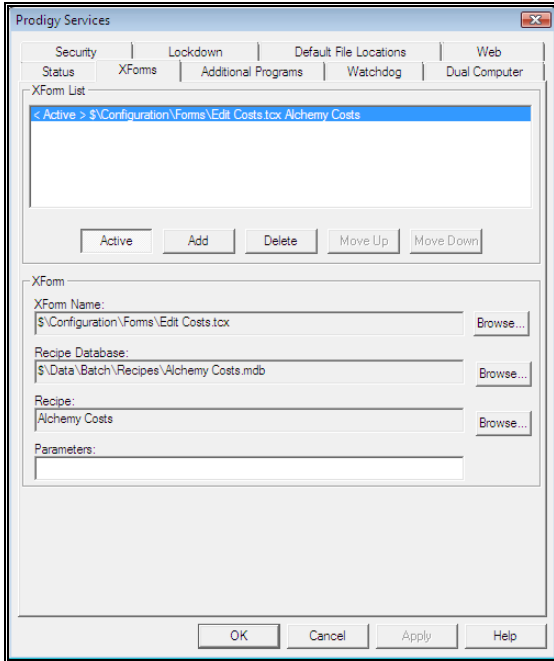
Software Controllers

If you select **Software Controllers** you will also have the option to select **New (With Auto Tuning)**. This option should **always** be chosen as the new controller, as well as allowing auto-tuning, also performs much better and much more predictably than the older controller. The older controller is only offered as an option to maintain existing functionality in those systems that are already using it. In these older systems you should still think about switching to the new controller as soon as is practical. Note that existing **P**, **I** and **D** terms from an old controller will not give good control when plugged into the new controller – the terms used by the new controller are typically several factors of magnitude removed from those of the old. Either run an

auto-tune or find your own manual entries by trial and error. A guide to the new P, I and D terms is given in the appropriate section of this manual.

XForms Tab

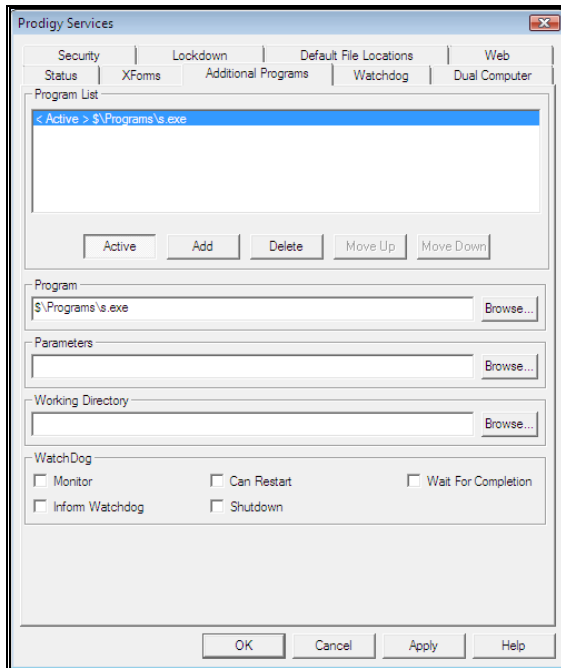
XForms can be written so that they require no user interaction, they simply pull values from a recipe and place them into signals. This is useful for providing start-up time initialisation of signals. Any XForms that are to be run in this way can be specified in the **XForms** tab of the control panel.



- **Active.**
If this button is depressed the current XForm is active and will be run when the services start. Clear this option to switch out XForms that are not required at present.
- **Add.**
Use this button to add new items to the list.
- **Delete.**
Use this option to remove items from the list.
- **Move Up.**
Use this to move an item further up the list. XForms are run in the order they appear in the list.
- **Move Down.**
Moves an item further down the list.
- **XForm Name.**
Use the **Browse...** button to select the XForm to be ran. The XForm should be designed so that it accesses only Recipe data and signals. If it accesses Batch data it will not run.
- **Recipe Database.**
Browse for the recipe database here.
- **Recipe.**
Browse the database (specified above) for the name of the recipe to be used.. (The XForm may have been written as a **Fixed Recipe** XForm. If so, this field is not required, but if given it will override the name of the fixed recipe).
- **Parameters.**
Specify any additional parameters required when this XForm is run (usually there will be none).

Additional Programs Tab

This tab allows the user to specify any additional programs that need to be started as part of the Prodigy Service. It is most likely that this list will be configured by Tascomp or your supplier and will run programs that they have supplied. It is not appropriate to start up some other third party software, such as Excel™, as part of the Prodigy Service.



- **Active.**
If an entry in the Additional Programs list is Active it will be started by the services. If it is Inactive it will not be started. This allows programs to be switched on or off depending on specific circumstances.
- **Add.**
Adds a new entry into the list.
- **Delete.**
Deletes the currently selected entry from the list.
- **Move Up / Move Down.**
The order in which the programs are started may be important. Use these buttons to move the currently selected item up or down the list.
- **Program.**
This is the name of the program that is to be run. Specify the path as fully as possible to minimise potential problems. Use the **Browse** button to locate the required program.
- **Parameters.**
Enter any parameters required by the program that you have specified.
- **Monitor.**
This indicates whether or not Prodigy should monitor this program to see if it stops running. This should only be used with programs that are meant to run permanently, and not for programs that run, perform a certain action or actions and then stop.
- **Can Restart.**
If a program is being monitored and it stops, the Service will restart the program if the **Can Restart** button has been checked. This option should only be used with programs that are designed to run continuously. For instance the **S** program used in our example above would not be a candidate, as it would set BATCH to 0 again when it starts, with potentially catastrophic results.
- **Wait For Completion.**
If this box is checked then the next program in the list will not be started until this one has terminated. If this box is not checked then the next program can be started in parallel with this one.

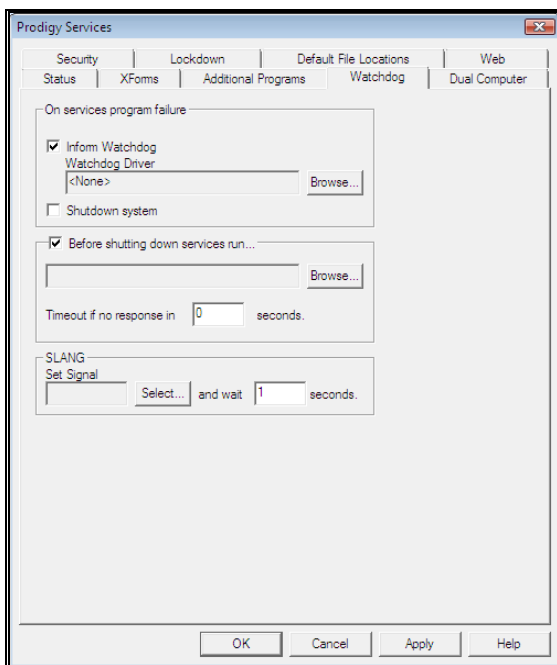
*Certain combinations of the above three options are not allowed. For instance, you can only use **Can Restart** if you have used **Monitor** and you cannot use **Wait For Completion** if you have used **Monitor**.*

- **Inform Watchdog.**
If this box is checked then if the program dies a message is sent to the watchdog driver (see **Watchdog** tab). This driver will most likely cause the machine to be restarted.
- **Shutdown.**
If this box is checked and the program dies then the Prodigy services will be stopped and restarted. The machine will not be rebooted, though.

Watchdog Tab

A watchdog is a piece of hardware that constantly monitors for system activity and if activity stops it performs a system restart. It ensures that the system does not suffer a crash or CPU freeze and not realise it.

The watchdog tab enables you to specify that if some part of Prodigy fails that the watchdog is informed and allowed to perform a system restart. If no watchdog is present you can still use this tab to configure what actions are taken if a part of Prodigy fails. The only situation that cannot be covered without watchdog hardware is operating system crash or CPU freeze.



The features of the Watchdog tab are as follows.

- **Inform Watchdog.**
If this box is checked then if a service or driver dies the watchdog driver will be informed and will cause the PC to be rebooted. If the relevant **Inform Watchdog** box has been checked on the **Additional Programs** tab, the watchdog will also be notified if an additional program fails.
- **Watchdog Driver.**
Select the Watchdog Diver from the list of drivers presented when the **Browse...** button is pressed. If no driver is selected, the **Inform Watchdog** check box will have no effect in the event of a program failure.
- **Shutdown System.**
If this box is checked the Prodigy services will be stopped in the event of a services, driver or (optionally) additional program failure.
- **Before shutting down Prodigy run...**
This allows the user to specify the name of a “controlled stop” program that will be run prior to shutting down the Prodigy services and (optionally) rebooting the machine. This program will be

custom made for a particular application and will close down the process in a safe and controlled manner.

Time out if no response in.

It is possible that the controlled shutdown program may itself fail, or may not be able to operate because some essential part of the services has died. In this case the services (or watchdog driver) will still cause the machine to reboot after the number of seconds specified here. Be sure to enter a number that is sufficiently large to allow the controlled shutdown procedure to finish, but is not so large that the system waits too long before rebooting.

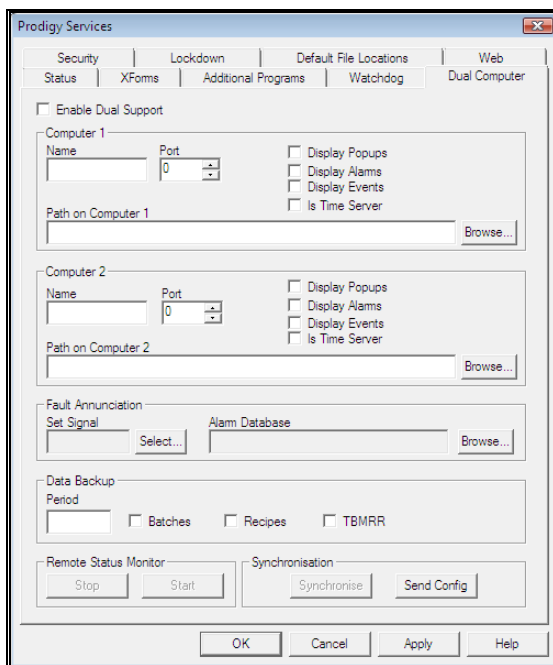
- **SLANG.**

Before stopping the services a SLANG program can be executed to perform some controlled shutdown procedure. The SLANG program is triggered by setting the signal specified here (so it should be running but waiting for this signal to be set before it performs any actions).

Remember that if part of Prodigy has died it may not be possible for the SLANG program to execute, or to perform all of the actions it is programmed with.

The “**and wait**” field specifies how long to wait after triggering the signal before initiating the shutdown.

Dual Computer Tab



To enable Dual Computer Support, check the *Enable Dual Support* box at the top of this display, and then configure the support by filling in the fields below.

There are three groups of configuration parameters on this display, for *Computer 1*, *Computer 2* and *Fault Annunciation*. The options used in *Computer 1* and *Computer 2* are the same, but relate to the respective computers.

- **Computer 1 and Computer 2.**

Name.

Enter the name of the computer. If this box is left blank, the name will be automatically filled in when you successfully browse the path of the data (see below).

Port.

Enter the number of the TCP/IP port on which communication will take place. The port number must be the same for both computers.

Note that if you are using the *Prodigy Network2 Driver* with *Prodigy Interconnect* or *Remote Workstation* the port numbers used there must not duplicate the port numbers used for the Dual Computer Support.

Path on computer n.

Enter the path of the Prodigy directory. This path must be obtained by going via Network Neighbourhood so you must be sure to share at least some part of the path to the Prodigy directory.

Display Popups.

If this box is checked then any fault conditions that are detected by Remote Status Monitor will be reported by popping up a dialog containing the relevant error message.

Display Alarms.

If this box is checked then any fault conditions that are detected by Remote Status Monitor will be reported by generating an alarm. This is achieved by setting a digital signal to 1 (see Fault Annunciation below).

Display Events.

If this box is checked then any fault conditions that are detected by Remote Status Monitor will be reported by generating an event that is displayed in the Alarm Display window.

Time Server.

One of the computers can be designated as being the time server. This means that it will periodically send the current date and time to the other computer. If the other computer finds that there is a difference between the time it has been sent and its own clock, it will correct its own clock. This will keep both computers to the exact same time.

- **Fault Annunciation.**

Set Signal.

Choose a signal that will be set to 1 when the Remote Status Monitor detects a fault condition. This signal should be a digital and you might want to set an alarm strategy that generates an alarm when the signal value goes to 1. Alternatively, you could use this signal to drive a piece of *SLANG* code or cause some display action on a *Display*. When the fault condition is cleared, this signal is set back to 0.

Alarm Database.

Browse for the Alarm database that you want any alarms and/or event texts generated by Remote Status Monitor to be recorded into. This should be the same database that you have specified for any alarm recording that is taking place as part of your alarm strategies.

- **Data Backup.**

Period.

Choose the period over which data will be backed up between the two computers that are been used for the Dual Computer Support.

Batches, Recipes and TBMRR.

Select what data you want to be backed up between each of the computers per every time period.

- **Synchronisation.**

Synchronise.

This option is only available when the services are not running. It copies the configuration and data from the other computer to this computer. A dialog is displayed that allows you to configure the transfer (see later).

Send Config.

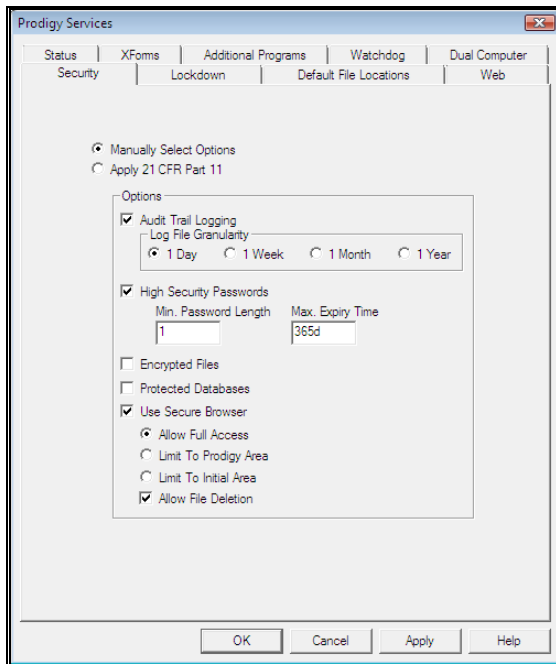
Pressing this button copies the Dual Computer Support configuration that you have on the screen to the other computer. This means that you can set up the configuration on one machine and copy it across to the other one. Then, when the services start up on the other computer, it will be pre-loaded with the correct Dual Support configuration.

- **Remote Status Monitor.**

There are two buttons here that allow you to stop and start the Remote Status Monitor. It is unlikely you will ever need to do this, but the facility is here nonetheless.

Whenever you save the Dual Support configuration data, the Remote Status Monitor re-reads the configuration.

Security Tab



The security tab allows you to select a number of security options to be applied throughout Prodigy.

- **Manually Select Options.**

This allows you to manually select and configure the individual security options.

- **Apply 21 CFR Part 11.**

Selecting this option will automatically select and configure those security options necessary to implement the recommendations of 21 CFR Part 11. Specifically these are:

Audit trail logging.

High security passwords.

The **minimum password length** is set to 6 characters and the **maximum expiry time** is set to 180 days.

Encrypted files.

Protected databases.

Secure file browser

See the entries below for more details on what they provide.

Applying CFR is a one-shot option. Once applied it cannot be removed. All encrypted files will remain encrypted forever, and similarly for protected databases. Be sure **before** you do it that you really want to apply this option.

- **Audit Trail Logging.**

When this option is enabled various security and traceability related actions are logged to a database. Such actions include users logging in and out; program invocation/termination; users obtaining privileges; failure to obtain privileges (wrong password give, etc); changes to user accounts; acceptance of alarms; changes to system configuration (Displays, signal database, BDR, etc. – in fact all configuration programs log who made changes with them).

The data recorded by the Audit Trail Logger may be viewed later using the **Audit Trial Viewer** (See separate entry in manual).

Log File Granularity.

This controls how much data goes into an individual database file – one day, one week, one month or one year. A new database is created at the beginning of each new period - every day; on Monday of each week; on the 1st of each month; or on 1st January each year. How you decide what to choose is quite arbitrary and may be based on how much data you expect to log. In general try to avoid a situation where the database file might become huge (containing thousands or tens of thousands of records).

- **High Security Passwords.**
Checking this box allows you to configure the global minimum for password length (this may be tailored – but only to a higher value – on an individual user basis) and also the global maximum for the password expiry period – again this can be tailored on an individual basis.
Checking this box also has the effect of enabling the **Security** button in the **User Configuration** program (see elsewhere in this manual). Note that when **CFR** is selected, these fields are automatically set to 6 characters and 180 days respectively.
- **Encrypted Files.**
Checking this box will cause all text based configuration and data files to be encrypted. The encryption process is irreversible and once encrypted all affected configuration and data files will be readable only through the Prodigy programs to which they relate.
The encryption is full 128 bit multi-cyclic encryption, and should resist even the most sophisticated and determined of would-be hackers.
This is a one-shot option and it cannot be reversed. The system will prompt to make sure that you want to perform these changes before it proceeds. Once you have confirmed it, the entire Prodigy area will be scanned for text-based files and they will be encrypted.
- **Protected Databases.**
Checking this box will cause all Access™ or SQL Server databases to be password protected. The password used will be hashed by the Prodigy system on an individual database basis. Once applied, there will be no way to open the database other than via a relevant Prodigy program.
As with Encrypted Files this is a one-shot option and it cannot be reversed. You will be prompted before these changes are made.
- **Use Secure File Browser.**
Checking this option means that throughout Prodigy whenever a file browser is used the standard Windows file browser will be replaced by Prodigy's own file browser. The Windows file browser allows too many options that are often best prevented – especially when applying CFR (for instance the ability to delete, rename and copy files; the ability to launch programs; the ability to browse to other areas of the computer). The Prodigy Secure File Browser can eliminate these security loopholes. See later in this manual for more details about the Secure File Browser.

Allow Full Access.

If this option is selected the Secure File Browser will still allow browsing to other areas of the computer and network (note though that with the Secure File Browser only mapped drives can be accessed, and not the entire Network Neighbourhood).

Limit To Prodigy Area.

If this option is selected the Secure File Browser will not allow browsing outside of the Prodigy root directory. Any directory that is a part of the Prodigy area can be accessed.

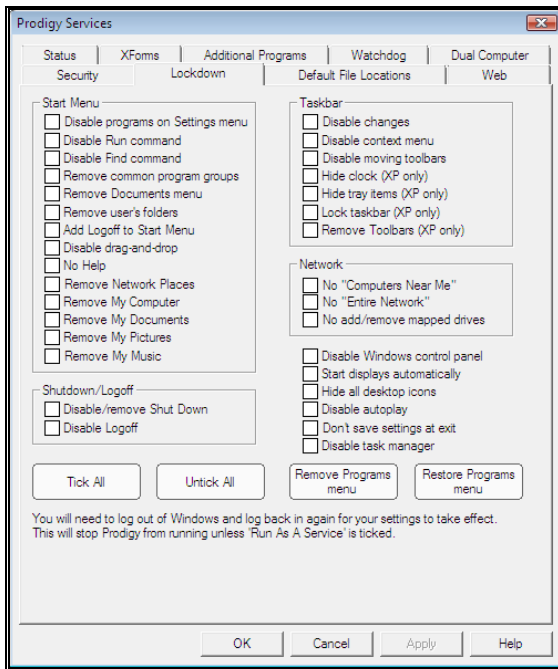
Limit To Initial Area.

If this option is chosen then only the area into which Prodigy has launched the Secure File Browser and any of its sub-directories can be accessed. This is the most secure option, and the circumstances under which a user (rather than a configurer) would need to browse elsewhere are very limited.

Allow File Deletion.

If this option is selected file deletion is never allowed from within the Secure File Browser. Even if the option is deleted, the circumstances under which file deletion is allowed are limited to those situations that constitute configuration.

Lockdown Tab

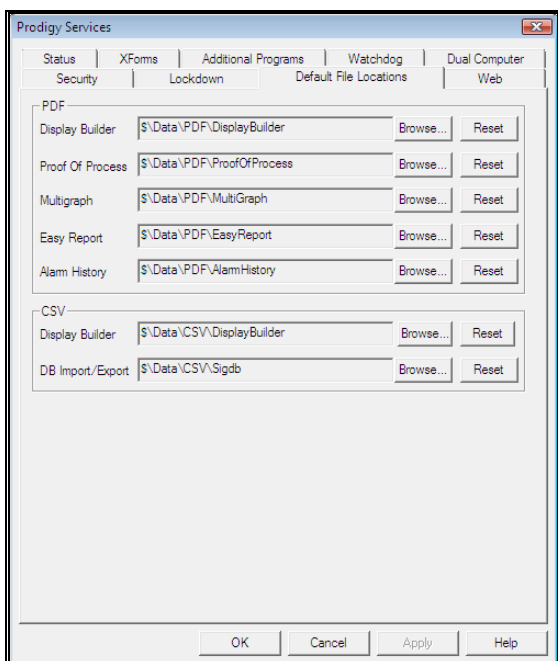


The lockdown tab enables you to quickly and easily configure the computer desktop (not Prodigy) so that access is given only to those aspects of the desktop that you want to allow. This will prevent the users of the Prodigy system from gaining access to Explorer, system tools, games, the internet, network resources and so on. This kind of lockdown is recommended in many application areas where you want to prevent “tampering” with the system, either intentional or otherwise. The options on the dialog are hopefully self-explanatory.

N.B. The option **Remove Programs menu** will only remove programs menu items that are for *this user* only. Options that are for *all users* will remain. The menu is copied to a backup area and can be restore as required by the **Restore Programs menu** button.

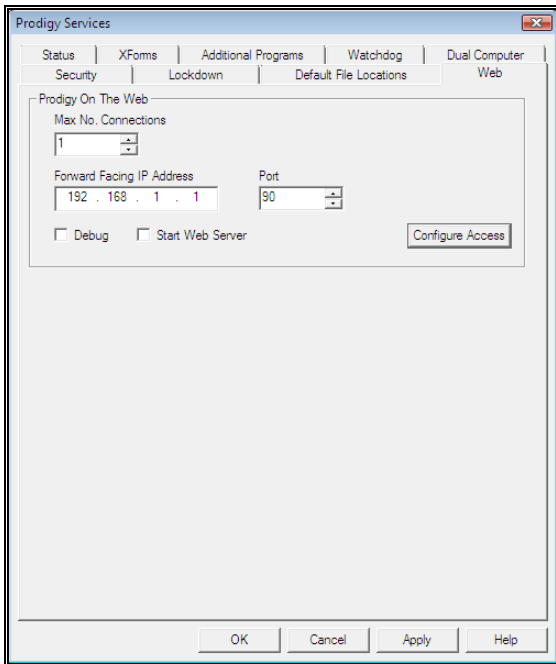
Default File Locations Tab

This allows you to specify the default location of the output from various Prodigy programs. In particular it is mainly for the output location of PDF file output from most of the reporting programs found within Prodigy.



Click on the relevant **Browse** button to select a location for that particular form of output. Hit the **Reset** button to restore a location to its system default.

Web Tab



The web tab allows you to set up Prodigy so that displays can be accessed through Internet Explorer. Please refer to the Prodigy Web Viewer section for information on configuring the Web tab.

Application Monitoring

When the processes that are started by Prodigy Services are running and one of them dies, the Service will notice this only if **Monitor** has been selected, or if the process was a driver or one of the core processes (all drivers and core processes are monitored).

If the service has been told that it can restart the process (which is true for some of the core processes, all drivers and those additional programs that have had **Can Restart** checked) then the service will attempt to restart the process. If the process dies six times in quick succession (i.e. in less than 5 minutes) the service will stop attempting to restart it and will issue an error dialog.

If a process dies that the service is not allowed to restart, it will issue an error dialog.

The built in **Watchdog** facility will act in the following circumstances

- If an additional program dies and either the **Inform Watchdog** or **Shutdown** boxes have been checked. If **Inform Watchdog** has been checked the services will notify the relevant **Watchdog Driver** (see **Watchdog**, below). If **Shutdown** is checked the services will be restarted.
- If a driver or a core services program dies the services Watchdog will act in the manner specified on the Watchdog tab.

If programs die the user should try to rectify the problem, which may involve calling their Prodigy supplier.

Actions Of The Watchdog

In general, if a driver dies it can be restarted with little overall effect on the system. If any of the core services die, though, the system must be restarted. This can be done manually by the operators, but there are times when it may be necessary for the system to restart itself automatically, for instance when it is left running

unattended. The automatic handling of service restarts and even computer reboots is managed by the **Watchdog** tab.

Watchdog Driver

It is possible that any application may stop running, even the operating system. Because of this Prodigy may be configured to use a Watchdog card and driver. The card needs to be reset by the software driver at regular intervals so if something goes wrong with the PC and it crashes, the card will not get reset. This causes the card to reboot the computer. In addition, Prodigy can be configured so that if any of the services or drivers die the watchdog driver is informed and it will stop resetting the watchdog card, causing the PC to be rebooted. Prodigy may also be configured to run a **Controlled Stop** routine prior to restarting the services or rebooting the PC.

Deferred Reboot

It is possible that something may cause a driver or service to die immediately, during the service start-up procedure. This would cause the watchdog to reboot the system immediately after start-up, allowing no time to attempt to rectify the fault! Because of this the watchdog will not reboot the PC until one minute after the services have started. The best thing to do during this minute is to turn off the watchdog features until the problem is rectified.

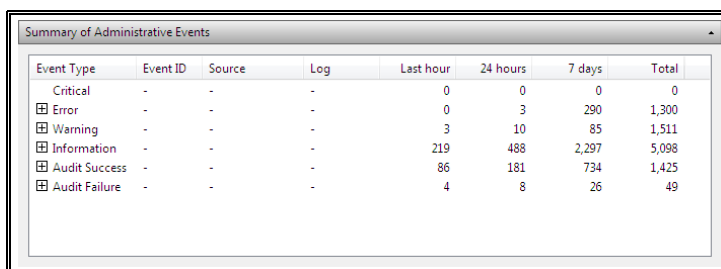
Stopping the Services

If the services are stopped, either from the Windows Control Panel or from Prodigy, **ALL** Prodigy programs that are running will be stopped. Note that it is very important to save any work in progress, be it in **Display Builder**, Database Configuration, BDR Configuration or anywhere else within Prodigy *before* stopping the services. No Prodigy program, other than **User Configuration** can run when the Services are stopped.

Error Logging

Any errors that occur within the Prodigy Services are reported via the Windows Event Logging mechanism. Errors may occur when a program cannot be started or when a program dies and in such instances a message is written to the event log. If the service attempts to restart the process an informative message is written to the event log. When an error occurs the Current Status in the Prodigy Service Control Panel will reflect this fact. The error log can be viewed by using the **Event Viewer** program, which is a standard Windows Vista facility. It can be invoked from the Control Panel under **Administrative Tools (Common)/Event Viewer**.

Once Event Viewer is open it is possible to view any Events that have occurred, a summary is shown like the one below and you can navigate to specific types of events, e.g. Errors, Warnings, Information etc. Any Errors caused by the Prodigy system can be observed by double clicking **Error** and then selecting **Prodigy System Service** this will give a list of all the errors



Event Type	Event ID	Source	Log	Last hour	24 hours	7 days	Total
Critical	-	-	-	0	0	0	0
Error	-	-	-	0	3	290	1,300
Warning	-	-	-	3	10	85	1,511
Information	-	-	-	219	488	2,297	5,098
Audit Success	-	-	-	86	181	734	1,425
Audit Failure	-	-	-	4	8	26	49

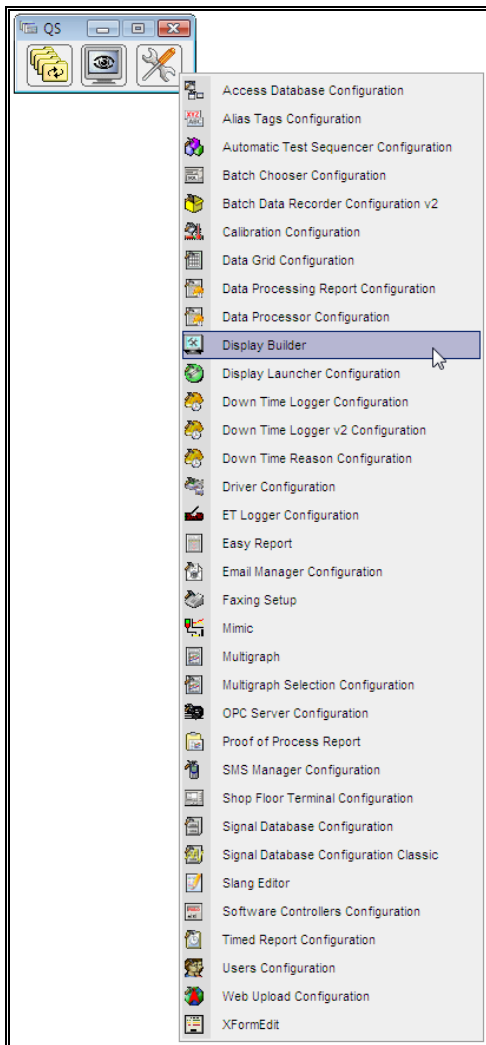
Use **Help** from within the event viewer for more information.

Quick Start

Quick Start



Quick Start is a small floating toolbar that gives access to all of the user applications within prodigy. It consists of three icon buttons representing **Configuration**, **System** and **Tools**. Clicking on one of the buttons reveals a menu of programs available within that category. It replaces the access to programs that used to be provided by the Windows Start button.



The Quick Start bar itself does not handle anything to do with security – access is given to all Prodigy programs - but of course many programs have built-in privilege requirements (e.g. any of the configuration programs). Access to the Quick Start program itself should be protected to prevent operators gaining access to programs that they should have no access to.

The Quick Start bar is **always on top** and is deliberately small and compact so that it can be left on your screen without obscuring too much of the display.

How Security Works

Prodigy Security works by allowing only authorised users to have access to the different features of a Prodigy system. This is achieved by maintaining a database of users and assigning a list of privileges to each user. These privileges determine which functions the user can access within Prodigy.

When a Prodigy system is running there may be a “logged on user”. Whenever a facility requiring privileges is accessed the system checks the required privilege against the privileges held by the logged on user. If the user holds the required privilege, access is granted. If the logged on user does not hold the privilege the system prompts for a user ID and password. The user can then give the ID and password of someone who does hold the necessary privilege. This user does not become the logged on user, but the system is aware that they have overridden the logged on user.

This methodology allows the system to be configured so that a low-level user ID can be created that allows automatic access to a number of mundane facilities while requiring a higher privileged ID to be entered for more sensitive facilities. This low level ID is known as the **Default User** and when **Prodigy Services** start up they automatically log this user on.

By giving the Default User no privileges whatsoever every action that requires privileges will require the user to enter an ID and password. This gives the most secure system, but can prove rather tiresome in use.

When *Security* is implemented most programs within Prodigy log the fact that they have been run (and by whom) to the **Audit Trail**, including details of the logged-on and overridden user. Any programs that are used to modify a configuration (e.g. *Signal Database Configuration*, *Display Builder*) will write a record to the Audit Trail indicating what was changed, when and who the user was that made the changes.

User Management

The **Users** program allows user IDs to be created and privileges assigned to these users. It takes the form of a dialog, shown below.

The screenshot shows the 'Users' dialog box with the following data:

Log on name	First Name(s)	Last Name	SMS Number	Act	Email Address	Act
Default User	Default	User		N		N
Plant Manager	Gordon	Smith	+447798345123	Y	gordon.smith@g...	Y

Buttons: Add, Delete, Copy, Security

Log On:
Log on Name: Plant Manager, First Name(s): Gordon, Last name: Smith, Password: [empty]

SMS Alarms:
Telephone Number: +447798345123, SMS Alarm Sets: Active, Restart

Email Alarms:
Address: gordon.smith@goldfromlead.com, Email Alarm Sets: Active, Restart

Web Viewing:
Initial Display: [empty], Browse...

Privileges:

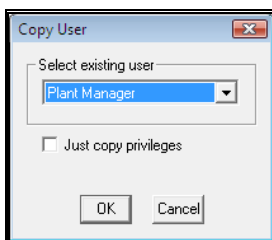
Privilege	Reason	Allowed
Accept Alarms	No	Yes
Alter Advanced Alarm Properties	No	Yes
Alter Alarm Properties	No	Yes
Configure Access Databases	No	Yes
Configure Alarm Display	No	Yes
Configure Alias Tags	No	Yes
Configure Archive	No	Yes
Configure ATS	No	Yes

Buttons: New..., Delete, Requires Reason, Has Privilege, Set all, Clear all, OK, Cancel, Help

User Details

At the top of the dialog is a list showing all of the users in the database. Click on an entry in the list to display that user's details.

- **Log on name.**
This is a mnemonic that the user will use to log on. It can be anything, but a good idea is to use initials, names or nicknames.
- **First Name(s).**
The user's first name or names. Along with the Last name this field may be used in reports to identify the user correctly rather than by their log on name.
- **Last name.**
The user's last name. This may be used in reports to identify the user, as an alternative to using their log on name.
- **Password.**
The user's password. The user must enter this password every time they log on. If the user forgets their password, enter a new one here. After setting a user's password here, the user will be prompted to change their password the first time that they log in with the password you supplied. This is a security measure to ensure that not even the Prodigy administrator knows the passwords of other users.
Caution. Be careful not to forget the password that allows you to access the Users program. If you do there will be no way to gain access to the program other than by contacting your supplier.
- **SMS Alarms.**
See the relevant section of the manual.
- **Email Alarms.**
See the relevant section of the manual.
- **Web Viewing.**
Defines the initial display that the user will see when they log onto the prodigy system through a configured web server system.
- **Add.**
This button allows you to create a new user ID. Initially the system will create an entry called New User. Change the Log on name, password and privileges and (optionally) enter the first and last names of the new user. Once you have finished modifying the details for this user and perform some other action (add another user, select a different user from the list, etc) you will no longer be able to modify the Log on name. If a Log on name requires changing the only way to do it is to add a new entry with the correct name and then delete the old one. Use the **Copy** button to facilitate this.
- **Delete.**
Deletes the currently selected user from the database.
- **Copy.**
Allows you to copy all of the details from another user to this user. Pressing the Copy button presents a dialog that allows you to select the user whose details you wish to copy.



It is most likely that you will only want to copy the privileges from the existing user, and the system provides a check box to allow for this.

- **Security.**
This option is only available if "High Security Passwords" is checked on the Security Tab of the Prodigy Control Panel. Clicking this button brings up the advanced Security dialog, described in more detail below.

Security

Clicking the Security button displays the dialog shown below.

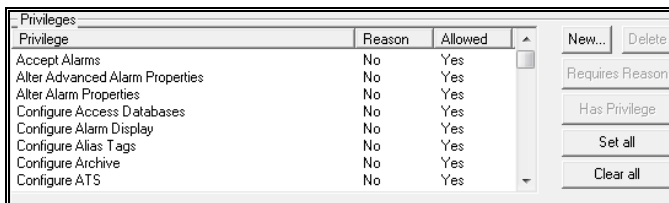


This dialog allows you to configure, on an individual user basis, some advanced security features surrounding user passwords.

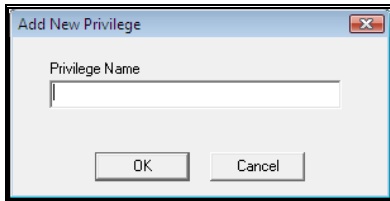
- Min Password Length.**
 This is the minimum number of characters that a user's password must consist of. If this value is changed then all users will be prompted to change their passwords the next time that they log in. The value for minimum password length set here must be no less than the global value set in the Security tab of the Prodigy Control Panel.
- ID is blocked.**
 Checking this box will block the user's ID. This means that the user can no longer access their account until the box is cleared. The ID may be blocked manually (by using this dialog) or it may (more likely) be blocked automatically by the system. Occasions when this would occur are when someone has tried to log into the account 5 consecutive times without success or when the password expires and the user does not provide a new one within the expiry period. Unchecking this box will reactivate a blocked ID. The user will be asked to provide a new password the next time that they log on.
- Force change password at next login.**
 Checking this box will mean that the user must change their password the next time that they log in. This option is forced anyway if the user's password is set via the User Configuration screen or if the minimum password length is changed. You cannot uncheck this box, the user **MUST** change their password.
- Password expires.**
 If this box is checked then passwords will expire. The expiry period must be set and must be no greater than the global value set in the Security tab of the Prodigy Control Panel. When the expiry period has passed the user will be prompted to change their password. If they do not, the ID will be blocked.

Privileges

Below the user details section is the Privileges section. This contains a list of all privileges that the system allows along with an indication of whether the currently selected user has those privileges. To change the status of a privilege double click on the relevant privilege in the list or press the **Has Privilege** button. The indicator to the right of the list will change from "Yes" to "No" or vice-versa.



- New...**
 Pressing this button allows you to create a new Privilege. Prodigy has a number of pre-existing privileges that the various facilities within Prodigy will check. You can enter privileges of your own to limit access to certain XForms, Menu Bar items, Programs etc. Pressing the button displays the following dialog.



Enter the name of the new privilege you wish to create in the box and press OK.

- **Delete.**
Deletes the currently selected privilege, if it is a user-defined privilege. Prodigy system-defined privileges cannot be deleted.
- **Has Privilege.**
Press this to set or clear the current privilege for the currently selected user. Alternatively the privilege can be double-clicked in the list.
- **Requires Reason.**
Press this button to set or clear the **Requires Reason** flag for the selected privilege. If this is set and a user attempts to gain the privilege, they will be asked to provide a reason as to why they want that privilege.
Note that this option is applied to the privilege for **all** users, it is not on a per-user basis.
- **Set All.**
Gives the currently selected user all of the privileges in the list.
- **Clear All.**
Removes all of the privileges from the currently selected user.

Take a look at the section in this manual about the **Security Tab** of the **Prodigy Control Panel** for information about these advanced security options.

The Default User

Whenever Prodigy is started the system attempts to set the currently logged in user to “Default User”. If this user ID does not exist then the system assumes you are operating without the facility of **Users**. This is perfectly acceptable and may be the way in which you wish the system to operate. Limited security can still be provided by means of passwords on individual items, but the most flexible method is to use **Privileges** through the **User Configuration** and **User Database**.

The reason behind having a Default User is that you can assign a number of privileges to it that enable various functions to be performed without a user having to log on. In fact, it will allow users who do not have a specific ID of their own to perform certain tasks. Any tasks requiring a privilege not assigned to the Default User will require the entry of an ID and password of a user having the required privilege.

The very first time you enter the Users program the system checks to see if a Default User account exists. If one does not exist it will create one for you and assign all access privileges to it. You should configure Default User to have only those privileges that you want it to have.

Make sure that there is always at least one User ID with access to the Users program, otherwise you will not be able to get back into the Users configuration program.

Logging On

Suppose you have created a security system whereby only the account **Manager** can access the various configuration programs (Signal Database, Display Builder Edit, Shop Floor Terminal, DTL, BDR, etc). Every time that user tries to access one of those systems they will have to enter their User ID (**Manager**) and password. This could prove tedious in a lengthy configuration session. Prodigy therefore allows users to **Logon**.

The Logon button is an optional part of the Menu Bar. When enabled it shows the ID of the currently logged on user. Clicking on the Logon button will display the Logon dialog.

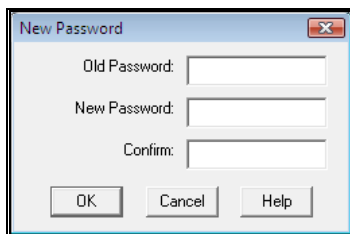


The user should enter their ID and password and then press OK. The Logon button on the Menu Bar will change to show that they are now the logged on user. Now whenever the user tries to access a part of the system it will check the required privilege against the privileges that user holds.

If the user gets their password wrong, they will have the option to try again. After 5 successive failures this user ID will be blocked (locked out).

In some circumstances as soon as the user has entered their Username the dialog will prompt them to change their password. Such situations include when the password has expired and when the password was set up by the Prodigy administrator. Also, the user can change their password at any time by clicking the **Change...** button after entering their Username.

The Change Password dialog is as shown below:



The user must enter their old (existing) password, the new password and the confirmation of the new password. If all these items validate successfully, their password will be changed. If they get their old password wrong, they will have the option to try again. After 5 successive failures the user ID will be blocked (locked out).

Logging Off

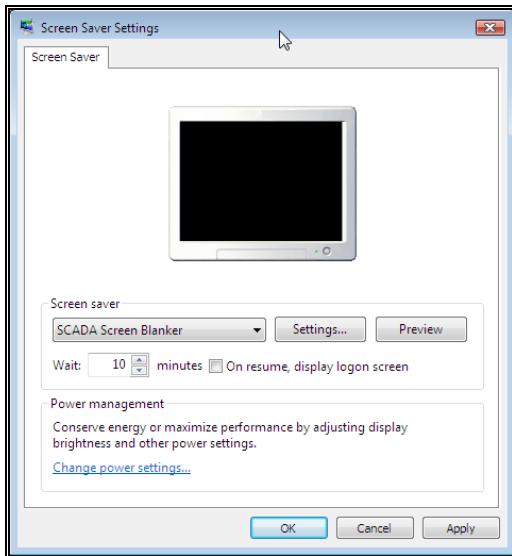
When the user has finished working at the terminal they should use the Logon dialog to **Logoff** (press the relevant button on the Logon dialog). Failure to do so will mean that anyone else who comes to the PC will be able to access those parts of the system that the current logged on user can access. Further more, the change log will show that any changes to the system were made by the current logged on user ID, which may not be the person who actually made the changes! Automatic Log Off can be achieved using the Prodigy Screen Saver and SLANG can be configured to perform periodic Log Offs.

Prodigy Screen Saver

Screen Saver & Auto Log-off

As an added security feature you can have Prodigy automatically log a specified user on after a certain length of inactivity. This means that if a Manager or some other user with extra privileges is logged in, and they leave the computer unattended for some reason, Prodigy will revert to the specified user ID so that others cannot take advantage of this.

In order to use the facility you use the Prodigy Screen Blanker. To do this right click on the PC desktop and select **Personalize** from the menu, then select the **Screen Saver** option. The following window will appear.

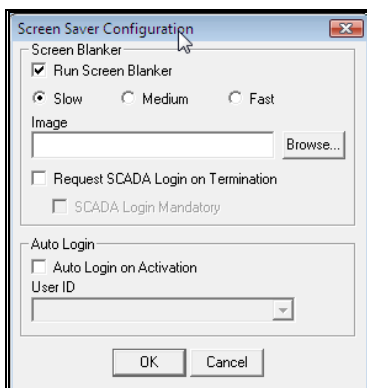


Select **SCADA Screen Blanker** as the Screen Saver and set the Wait to be the required number of minutes of inactivity. **DO NOT** check the **On resume, display logon screen** box as this moves the security outside of Prodigy's domain.

*Note that pressing the **Preview** button will run the screen saver with all of the options that have been selected for it, which means it might perform and auto login and might not let you terminate it without giving a valid Prodigy user ID and password. Be sure that you know a valid ID and password otherwise you could be stuck with the screen saver running and no way to get back out.*

Next click the **Settings...** button to display the configuration dialog for the Prodigy Screen Blanker.

The Screen Blanker moves up to 3 bitmap images around the screen. One of these images is provided by Tascomp, one is provided by the vendor and one by the user. They move around the screen on a black background at a rate determined in the configuration. Prodigy either displays this screen blanker, performs an auto log on, or both.



- **Run Screen Blanker.**
Check this box if you want the actual screen saver to come on. If this is not checked, Prodigy can perform the Auto Log On feature, but without actually displaying a screen saver. Select the speed at which you want the screen blanker to run.
- **Image.**
Browse for a bitmap that you want to be part of the screen blanker. The bitmap must be a 256 colour (or less) bitmap and for performance reasons it should not be too big (up to 20% of screen size should be OK). The colour of the pixel in the top left-hand corner of the bitmap image is taken as being the transparent colour.
- **Request SCADA Login on Termination.**
If this is checked, then when the screen saver stops (someone moves the mouse or presses a key) Prodigy will request the user ID and password of a valid Prodigy user. You can just press *Cancel* to proceed unless the following option is checked.
- **SCADA Login Mandatory.**
If this is checked then the screen saver will not terminate until a valid Prodigy user ID and password is entered.
- **Auto Login on Activation.**
If this box is checked Prodigy will automatically login the user that is selected in the box below. This user should normally be one that has few or no privileges.

Audit Trail

The Audit Trail within Prodigy is an optional feature, supplied as part of the FDA 21 CFR part 11 license option. If this license option has been purchased then Audit Trail can be turned on by checking the **Audit Trail Logging** box on the **Security** tab of the **Prodigy Control Panel**.

If this box is checked, **Audit Trail** logs the following events:

- Users logging on or off.
- Users gaining privileges because the logged in user already has the required privilege.
- Users gaining override privileges because the logged in user does not have the required privilege.
- Users entering the wrong ID/password whilst attempting to gain a privilege or log in.
- Lockout of user IDs, either manually or because of repeated password violations.
- Starting of a program.
- Ending of a program.
- Changes to configuration files.
- Changes to data values (e.g. signal values) via XForms, button presses on Displays and so on.
- Changes to user account details.
- Accepting of alarms.

A typical Audit Trail might show that:


- The services started (as part of this an automatic XForm ran and changed some signal values)
- A Menu Bar, Alarm Display and Display Builder was started.
- Fred logs in.
- The logged in user did not have the privilege to start Signal Database Configuration so an attempt was made to use the user ID and password for Joe to gain this privilege.
- Joe does not have this privilege either, so the id and password for Alan was tried next.
- Alan had this privilege, so the privilege was granted.
- The Signal Database Configuration program was started.
- Changes were made to the Signal database
- The Signal Database Configuration program ended.
- Fred changed his password
- Fred logs in again

The Audit Trail Viewer shown below lists all of these actions.

Time	Program	Event	Privilege	Logged In...	User	Action	Reason	Workstation
23-08-07 12:06:56	Stop SCADA	Started		Default User		Started		ck
23-08-07 12:06:56	Stop SCADA	Gained Privilege	Control Services	Default User		Gained privilege (is logged in user)		ck
23-08-07 12:07:00	Stop SCADA	Ended		Default User		Stopped (stopped services)		ck
23-08-07 12:07:04	Display Builder	Ended		Default User		Stopped		ck
23-08-07 12:08:42	SCADA Services	Started		Default User		Started		ck
23-08-07 12:09:08	*Form	Started		Default User		Started		ck
23-08-07 12:09:08	*Form	Opened		Default User		Opened %Configuration\FORMS\Edit Costs.tcx		ck
23-08-07 12:09:09	xformclassic	Changed Value		Default User		Changed value Recipe value COSTHG changed to 0.85 (from ???)		ck
23-08-07 12:09:09	xformclassic	Changed Value		Default User		Changed value Signal value COSTHG changed to 0.85 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTCU changed to 1.97 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTCU changed to 1.97 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Recipe value COSTPB changed to 0.35 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTPB changed to 0.35 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Recipe value COSTS changed to 5.85 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTS changed to 5.85 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Recipe value COSTLEC changed to 1.22 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTLEC changed to 1.22 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Recipe value COSTAU changed to 8.61 (from ???)		ck
23-08-07 12:09:10	xformclassic	Changed Value		Default User		Changed value Signal value COSTAU changed to 8.61 (from ???)		ck
23-08-07 12:09:11	xformclassic	Changed Value		Default User		Changed value Recipe values in %Data\Batch\Recipes\Alchemy Costs.mdb		ck
23-08-07 12:09:11	xformclassic	Changed Value		Default User		Changed value Signal values		ck
23-08-07 12:09:11	*Form	Ended		Default User		Stopped		ck
23-08-07 12:09:11	Menu Bar	Started		Default User		Started		ck
23-08-07 12:09:17	Alarm Display	Started		Default User		Started		ck
23-08-07 12:09:24	Display Builder	Started		Default User		Started		ck
23-08-07 12:09:25	Display Builder	Opened		Default User		Opened Mimic %Configuration\Welcome.tcm		ck
23-08-07 12:09:40	Log On	Logged On		Fred	fred	Logged on		ck
23-08-07 12:09:56	Signal Databas...	Illegal Access	Configure Signal Data...	Fred	joe	Failed to gain privilege		ck
23-08-07 12:10:01	Signal Databas...	Ended		Fred	fred	Stopped		ck
23-08-07 12:10:16	Signal Databas...	Gained Privilege	Configure Signal Data...	Fred	alan	Gained privilege (overrode logged in user "fred")		ck
23-08-07 12:10:16	Signal Databas...	Started		Fred	alan	Started		ck
23-08-07 12:11:02	Signal Databas...	Modified		Fred	alan	Modified : Saved Changes		ck
23-08-07 12:11:05	Signal Databas...	Modified		Fred	alan	Modified : Made Live		ck
23-08-07 12:11:10	Signal Databas...	Ended		Fred	alan	Stopped		ck
23-08-07 12:11:36	Log On	Changed Pas...		Fred	fred	Changed password		ck
23-08-07 12:11:42	Log On	Logged On		Fred	fred	Logged on		ck

Filtering

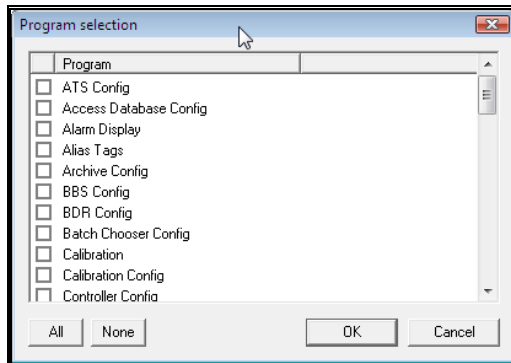
The Audit Trail Viewer is fairly straightforward to use. When it first runs up it shows all events that have

occurred today (since midnight). The display can then be **filtered** by pressing the  button to display the following screen.

- **Dates.**
Use the **From** and **Period** fields to specify the period that you want to look at.
- **Events.**
Check the required boxes for the types of event that you want to view records for.

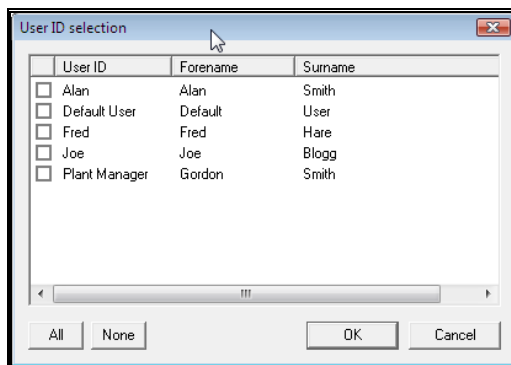
- **Programs.**

Enter the list of programs that you want to see events for, or leave blank to see all.



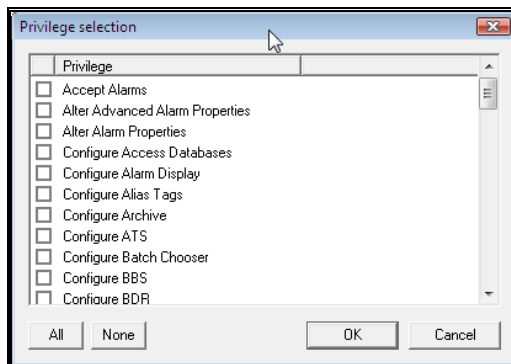
- **Users.**

Enter the list of users that you want to see events for, or leave blank to see all. Note that events will be shown for the users specified if they occur as the **Logged-In User** or the **User**.



- **Privileges.**

Enter the list of privileges that you want to see events for or leave blank to see all.



Sorting

The display can be sorted on any of the columns simply by clicking on the header of that column. Clicking the column a second time reverses the sort order on that column. Note that no matter which column the data is sorted on, the secondary sort is always on **Time**.

Printing

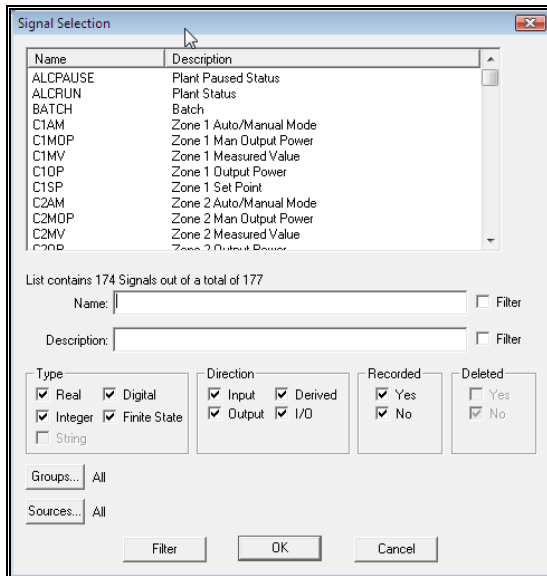
Clicking the Print button will print the current set of records (as selected by whatever filtering criteria have been applied). All of the records that have been selected will be printed, not just the ones that are visible on the screen (if there are more records than will fit on one screen).

SigScan

What is SigScan

SigScan is a dialog that is accessed from many parts of Prodigy, but mainly from within **Display Builder**. It is used whenever a signal tagname is required to be entered and provides a mechanism for allowing the user to *home in* on a signal based on a whole host of criteria.

Using SigScan

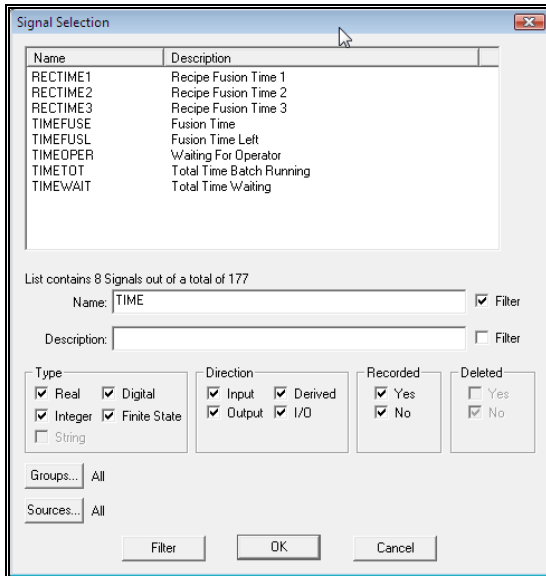


The dialog contains a list box that is split into two sections, the signal name and the signal description. Clicking on the heading “Name” causes the list to be sorted in ascending order of signal name (it is already sorted this way when it appears). Clicking it again causes it to be sorted in descending order. Similarly clicking on the heading “Description” causes the list to be sorted on either ascending or descending order of description.

The Signal Selection dialog allows the designer to perform certain filtering actions on the list of signals, to aid them in choosing the correct signal. There are various criteria by which signals may be filtered.

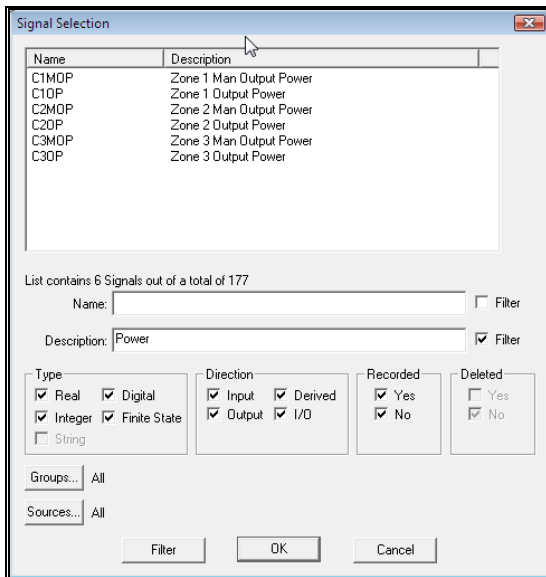
Name.

Enter a substring to search for and check the Filter box. The list will be refreshed to contain all signals whose name contains that substring.



Description.

Works exactly as for Name.



Filtering always occurs when a filter check box is checked or cleared. However, for both name and description, if you alter the substring text when filtering has already occurred the list will not be refreshed until you press the Filter button at the bottom of the dialog, or until you check or clear any of the other filters.

Type.

Five signal types are supported by Prodigy:

- Real
- Integer
- Digital

- Finite State
- String

Checking the relevant boxes causes only signals of that type to be displayed.

Digital and Finite State are both actually special cases of the Integer type. Finite State is a fixed range of integer values and each value in the range can have descriptive text associated with it, e.g. 0 to 2 with the text “Stopped”, “Slow”, and “Fast”. Further, Digital is in fact a special type of Finite State, having a range of 0 to 1 and text usually of the form “Off” and “On”.

Direction.

- Input means values are read into Prodigy
- Output means values are written out by Prodigy into plant devices
- I/O means values are both written and read
- A derived signal may be calculated from the values of other signals, or it may have its value set by a **Display Builder** object, for instance a slider or a button.

Checking the relevant boxes causes only signals of that direction to be displayed.

Recorded.

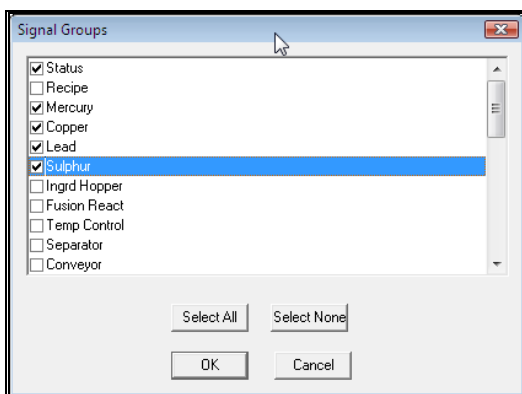
Some signals may have their values recorded to a history file on disk. Whether and when this occurs is defined in the Signal Database under “Recording Strategies”. Checking the relevant boxes causes only signals of that match to be displayed.

Deleted

Some signals may have been deleted from the signal database, but there might still be historical data that you want to be able to view. These two check boxes allow you to select signals that **have been** deleted or **have not been** deleted.

Groups.

Signals may be defined as being in any number of up to 32 different groups. Groups may be used to specify plant areas, equipment types, etc. and the Signal Chooser dialog can use this information to filter all signals in a given plant area for instance. The Group chooser is in the form of a Checked List Box.

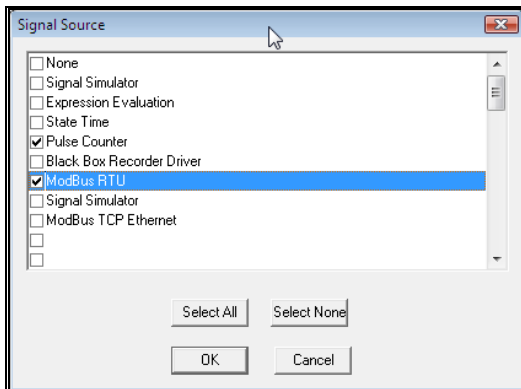


Check the box next to each group that you want to select signals from. In the example above, all signals relating to the current status **or** the raw ingredients (Mercury, Copper, Lead, or Sulphur) will be selected. **Note** that filtering takes place when OK is pressed on this dialog.

*When the signal database is defined, some signals may not be placed into any groups. Such signals will **always** be selected and there is no way to filter them out. If this is not desirable you could create a group called “None” and place all of these signals into that group.*

Sources.

A signals source can be one of many places. It may be from one of any number of hardware devices connected to the computer or it may be from some special software that generates signal values, for instance the *Evaluate* software calculates signal values from other signal values.



The Sources filter selects all signals that belong to any of the selected sources. **Note** that filtering takes place when OK is pressed on this dialog.

Filter Button.

This button is used to refresh the list, applying all filtering actions that have been specified. As filtering always occurs when a check box is checked or cleared, it is most likely that this button will be used when the text used to do signal name or description filtering is changed.

In addition to the filtering described above, the Signal Selection dialog can perform *partial name matching*. With the name filter checkbox cleared (i.e. **not** filtering on name) type in the first few characters of a signal name. The scroll box changes so that any signal name that begins with the characters you have typed is brought into view. If only one signal name begins with what you have typed, that signal is automatically selected in the scroll box, its name is placed in the name field and its description is placed in the description field.

When you have chosen the signal you want (by clicking on it in the list and pressing OK or by double clicking on it in the list) its name will be inserted into the relevant place on whichever dialog called up the Signal Chooser.

Sometimes when you bring up the Signal Chooser some of the items will be greyed out. This is because in some cases not all signal types are valid. For instance, when selecting a signal to be set by a slider it does not really make sense to choose a digital signal. The signal type "Digital" will not be checked and its checkbox will be inaccessible.

Signals Not In the Database

SigScan will sometimes show signals that are no longer in the signal database, but only where to do so is appropriate. When such signals are shown their description is always **No longer in signal database** and they do not respond to any of the filtering functions. This is because the system cannot know what type, direction etc. they are.

Alias Tags

What Are Alias Tags?

Suppose your Prodigy system is monitoring two areas of the plant, say two different production lines, A and B. Each line has signals for measured value (MVA and MVB) and setpoint (SPA and SPB). Suppose too that both of these areas are running batches, each with their own BDR configuration.

You configure up a trend of measured value against setpoint for each machine, so you have two trends. In the historical trend for line A you open a batch to look at the signal values for the duration of that batch. What if the batch was actually for line B? Yes, you would still see the values of the signals on line A, but for the duration of a batch running on line B. This could lead to some degree of confusion, and as previously stated, it means you need to configure up two trends. What if you have 20 production lines? It would be nice if you could configure up one trend and that trend could take the signal values from any of the 20 production lines. Well, **Alias Tags** allow you to do this.

How Do Alias Tags Work

You begin by defining which alias tags you are going to have in your system. These should be based around which real tags are duplicated in distinct areas of the plant. Only signals that record the same type of thing should be grouped together on an alias tag, otherwise this will lead to confusion in their use.

In the Prodigy Alchemy demonstration there are a group of signals measuring aspects of the Furnace.

Zone 1 Temperature Setpoint – C1SP

Zone 1 Temperature Measured Value – C1MV

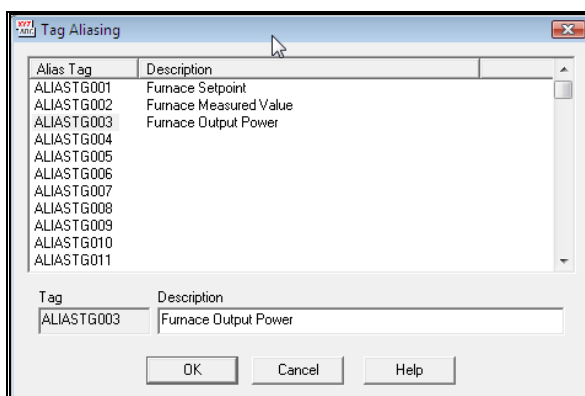
Zone 1 Percentage Power Output – C1OP

If there were 2 alchemy processes being monitored, these signals may be duplicated so that we would have C1SP1 and C1SP2, C1MV1 and C1MV2 and finally C1OP1 and C1OP2.

We could define alias tags for each of these three signals, one called **Furnace Setpoint**, one called **Furnace Measured Value** and one called **Furnace Output Power**. Each of the two real signals can then be mapped onto these alias tags.

Defining Alias Tags

You define Alias Tags by running the **Alias Tags** program. This is as follows

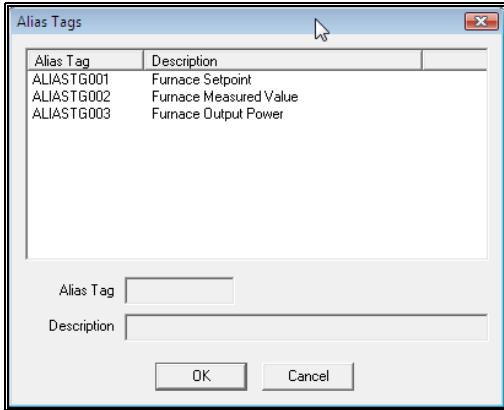


Alias tags have fixed names, this being **ALIASTG001** to **ALIASTG999**. This is 10 characters long and therefore cannot clash with real tags, whose length is limited to 8 characters. There are 999 alias tags available within a Prodigy system, but it is unlikely you will use anywhere near this amount.

The only alias tags that are available for use are the ones for which you have provided a description. Hence, in the illustration above we can see only 3 alias tags have been given a description, these matching our 3

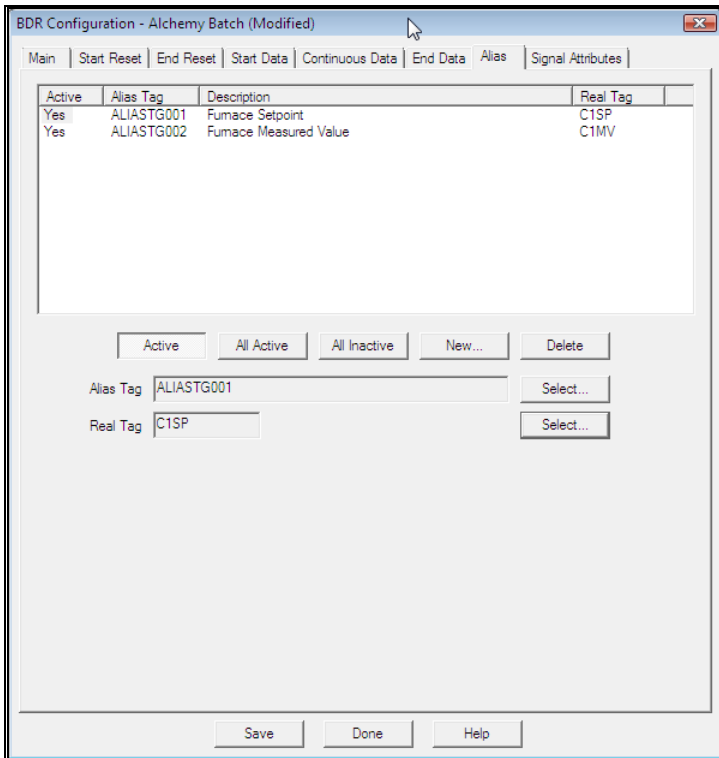
variables from the Alchemy demonstration. To give a tag a description, simply click on it in the list and then enter a description in the field on the dialog. To remove an alias tag, delete its description.

When it is time to select an alias tag for use (e.g. on a **Historic Trend**), a dialog is displayed that contains only those alias tags that are available, i.e. have had descriptions assigned to them.



Assigning Real Tags To Alias Tags

Real tags are assigned to alias tags by **BDR**. There is a tab on **BDR Configuration** that allows you to assign which real tags relate to which alias tags for a given batch definition. This tab is shown below.



When a batch terminates, BDR writes a record to the batch database, and as part of that record it writes which real tags relate to which alias tags for this batch.

A second BDR configuration would assign a different set of real tags to this same set of alias tags, and in that way when a batch record is read the system would know which real tags to use on the trend.

Using Alias Tags

Alias tags can be used in trends both within Display Builder and in a **Proof Of Process Report**. In both cases the alias tags are added to the trend by pressing the relevant button, in the same way as a real signal would be added.

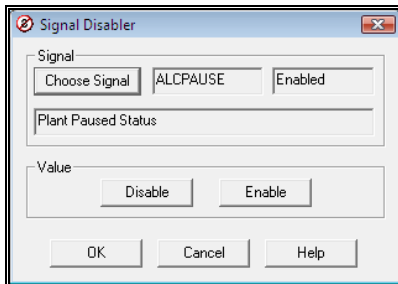
For trends within Display Builder, an alias tag will be excluded from the trend when it is in real-time mode, but once the operator switches to historical trends the system will request a batch name and substitute all alias tags for the real tags associated with that batch. These real signals will then be shown on the trend, along with their correct descriptions, scaling etc.

A **Proof Of Process Report** only works with historical data, so as part of its running process it performs the same alias-to-real tag substitution as historical trending.

Signal Disabler

Signal Disabler

When a signal is disabled it is treated as if it does not exist. Signals may be disabled when, for example, an area of plant is shut down for maintenance or a piece of hardware is faulty. The values coming back will be meaningless and may trigger alarms that are not valid, or may erroneously trigger other parts of the system. By disabling the signals any knock-on effects are minimised and the value of the signal is recorded as simply being disabled.



Choose the signal that is to be disabled and click the **Disable** button. To re-enable it select the signal and press the **Enable** button.

Signal disabling should be used with care and access to it should be limited as it is easy to forget (or not even realise) that a signal is disabled.

Display Replay

What Is Display Replay?

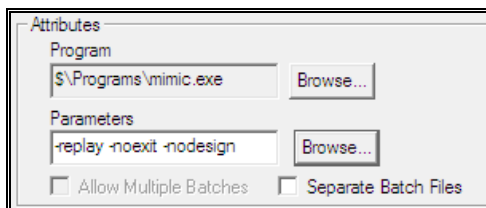
Display Replay allows you to step back in time and re-run a Display Builder display using data from historical recordings. Obviously only those objects on the display whose signals are recorded will respond in Replay mode. If you intend to make Display Replay a prime feature of your system it is best to bear this in mind when deciding which signals are to be recorded, and choose an appropriate recording strategy.

Using Display Replay

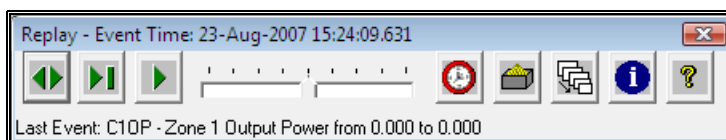
Display Replay is a licensable option, and if your license allows it you will see this button when in Edit mode.



You can run Replay directly from Edit mode by pressing this button. On a working system it is more likely that you will want to switch directly to Replay mode. This can be done by running Display Builder.exe with the parameter **-replay**, either from the **Menu Bar**, **SLANG** or via a button on a Display Builder display.



When in Replay Mode the Replay Bar appears. This is a toolbar as shown below.



The buttons on the bar are:

- **Change Direction.**
Pressing this button reverses the direction of replay (you can go backwards or forwards through a Replay).
- **Step.**
Advances by one event or *time period* depending on the mode.
- **Play.**
Continuously steps through the Replay either event by event or frame by frame.
- **Speed Control.**
In the central position the speed is 1x. Moving each graduation to the right doubles the speed (i.e. 2x, 4x and so on) and to the left halves the speed (i.e. 1/2, 1/4, and so on).
- **Time.**
Press this button to set the time from which Replay is to commence.
- **Batch.**
Press this button to select a batch. Replay will commence from the start of the batch.
- **Event Mode/Time Mode.**
This button defines the mode of Replay. When in Event Mode (button is as shown above) replay steps through one event at a time, regardless of how long may have elapsed between events. This is useful for seeing in very fine detail what sequence of events has occurred. What you see is not necessarily how you saw it when the Display was viewed "live" because when viewing it live several events might occur close together and be displayed in one refresh of the Display Builder display. Here they will be displayed one event at a time,.

The other mode is Time Mode and you switch to this by clicking the button, which will then appear as an hour glass with sand running through (below). In this mode time advances at a constant rate, determined by the refresh rate of the Display. Every event that has occurred within this time period is displayed in one redraw, so the Display will appear almost exactly as it did when first viewed.



Mode button when in Timed mode.

- **Show Information Window.**
- Clicking this button displays the Information Window, shown below. The window shows information about the current event as well as the previous and next 5 events. Note that the current event is the one that just happened, not the one that is about to happen (i.e. its effect has already been shown on the display). If you are single stepping and switch direction, you will notice that the **From** and **To** values swap and the effect on the display will be re-evaluated. By doing this you can replay over and over a single event.

Signal	Description	Time	From	To
LVLADISP	Acid Dispenser	23-Aug-2007 15:30:05.782	900.000	900.000
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:05.860	376.015	374.564
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:05.985	374.564	376.804
LVLABATH	Acid Bath Level	23-Aug-2007 15:30:06.110	125.000	125.000
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:06.110	376.804	379.825
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:06.235	379.825	382.745
LVLINGRD	Ingredients Hopper Level	23-Aug-2007 15:30:06.251	0.000	0.000
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:06.360	382.745	384.029
C1MV	Zone 1 Measured Value	23-Aug-2007 15:30:06.485	384.029	385.270

You exit Replay Mode by exiting the Replay Bar (click on the x). If you switch Displays during Replay mode you will be prompted as to whether or not you want to remain in Replay mode. The choice is yours.

What are XForms?

XForms are a dialog based mechanism whereby the operator can set signal values, *batch* file data values or both, either by entering the required values or by loading them in from pre-defined values held in *recipe* files. The dialogs are designed by the system implementer or the end-user and are called up by the system operators at the relevant points of the process cycle.

What are Recipes?

Recipes are a set of pre-defined values for certain process parameters. These may be things such as weights, volumes, times and temperatures that may be used as the setpoints of various plant devices. They may also be some numeric or text information that is simply written away to a batch file for future reference, but has no actual effect on the process.

A recipe is referenced by its name and any number of recipes may have values for the same process variables. For instance, a coffee vending machine may offer Black, White, Strong and With Sugar. The various recipes would set the quantity of:

- Milk to 0 (for black) or 1 (for white).
- Coffee to 1 (for standard) or 2 (for strong)
- Sugar to 0 (for no sugar) or 1 (for sugar).

Recipe information is accessed via XForms. It is likely that there will be an XForm that allows the recipes to be created or modified and a separate XForm that allows the operator to choose the recipe required for the current process.

What are Batches?

A process may either run continuously or may run for a set amount of time, for a set quantity of product or for a complete production cycle. In the latter three cases, the process may be said to be running for *batches*. Various process values may be recorded and discrete information written away that pertains to a given batch. Later, this information may be viewed either on Prodigy *Historical Trends* or as a printed report or on an XForm.

In Prodigy, the data that forms recipes or batch information is held in a Microsoft Access™ database, making it available for manipulation and reporting by a variety of Windows applications.

A Word About Opening Databases

In order for an XForm to work there must be a button on it that performs an **Open** action (there are exceptions, listed at the end of this topic). This button, when pressed, will prompt the user to select the record that he wants to view from the database table that is specified for this Open action. On selecting the required record, all fields on the XForm that are taken from that record are filled in and displayed. **Secondary Databases** (if any are specified) will then be opened too, using table and keying information specified in any Secondary Database Tables declared for this Open action. When using Data Grids this is a necessary action – data for Data Grids is taken from tables of repeating data that are keyed by the same field as the primary record. For example a standard batch database has the following tables and keys:

Table Name	Key Field 1	Key Field 2	Key Field 3	Key Field 4
Batches	Batch Name			
Details	Batch Name	Date/Time	Fractional Time	
Comments	Batch Name	Date/Time	Fractional Time	
Alarms	Batch Name	Signal Name	Date/Time	Fractional Time

You can see that all of the secondary tables are keyed by Batch Name, the primary key of the primary table.

In a standard XForm/Data Grid example, the Open button would open Batches as the Primary Database table and any one (or more) of Details, Comments and Alarms as Secondary Database Tables for any associated Data Grids.

A much more advanced usage of Secondary Databases is to open a completely different database, not simply a different table in the same database. This is achieved in exactly the same way, but instead of specifying the same database as the primary specify the required, different, database. All that is required is that the database is keyed by a value that appears on the XForm and which was read from the Primary Database. The facilities within the Secondary Database section of the Open action allow any field from the Primary database to be mapped to a key field of the Secondary Database. As an example of why you might want to do this, a batch may contain a field for the recipe that was used for that batch. By opening the Recipe database as a secondary database and mapping its key field to the recipe field from the batch record, the values from the recipe can be displayed on the XForm.

Details about how to configure a Primary Database open are given in the section about Buttons, specifically within the sub-section about Open action. Details on opening Secondary Databases are given in the section about the Data Grid object.

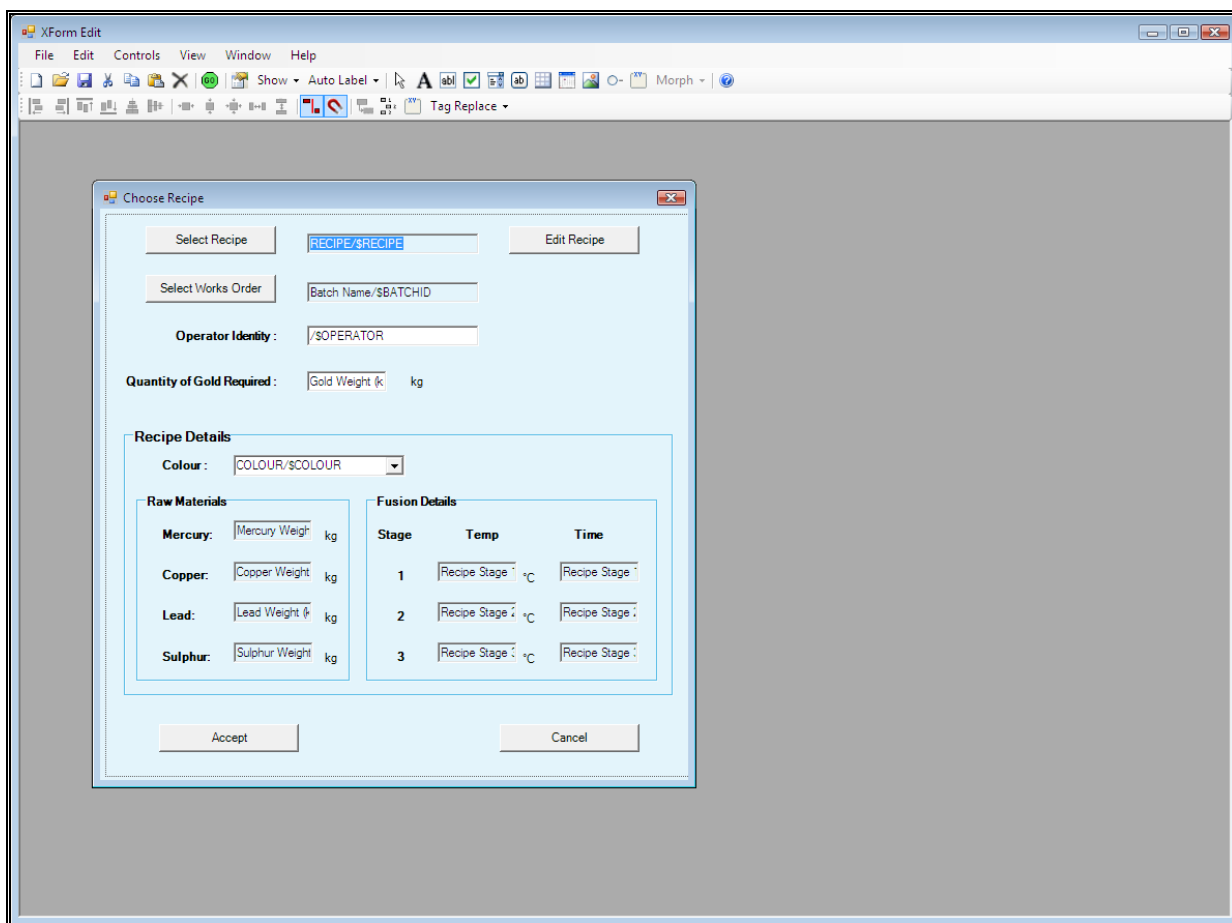
Exceptions

XForms *do not* need an Open button if they only have fields taken from signals, if they use a “fixed recipe” (which could equally be a fixed batch name or whatever the primary database key happens to represent) or if they are sub-forms. If all values are taken from signals then no secondary database can be opened. If a fixed recipe or sub-form is used then secondary databases can still be opened by mapping the relevant XForm field(s) to the key value(s) of the secondary table.

Launching XForm Edit

XForms are created or modified by using *XForm Edit* which can be found on the menu bar or within Quick Start.

The XForm Editor



The XForm editor is similar to Display Builder. You place objects in the same way, by selecting the relevant object type from the toolbar and then dragging a box on the display for the object to occupy. The XForm toolbar is shown below.

The XForm Toolbar



There are the standard New, Open, Save, Cut, Copy and Paste buttons. The other buttons' actions are described individually later.

Creating an XForm Dialog

To create an XForm press the new button, this will bring up a window that is your new XForm and also the Form Properties dialog.

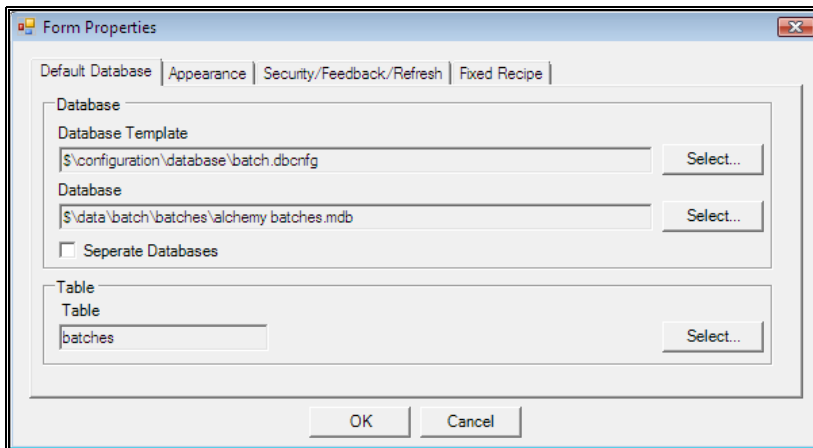


XForm Form Properties

The XForm's properties dialog appears when you start a new form; after this they can be accessed in two ways; by pressing the properties button (below) or by right clicking on the XForm's background.



Default Database



- **Database Template.**

The configuration file is used to create the empty database if it does not already exist – see **Access Database Configuration** elsewhere in this manual for further details. Click **Select...** next to the Database Template field to select the correct configuration file. (Note, there is a default Batch configuration file which will create a standard batch file).

- **Database.**

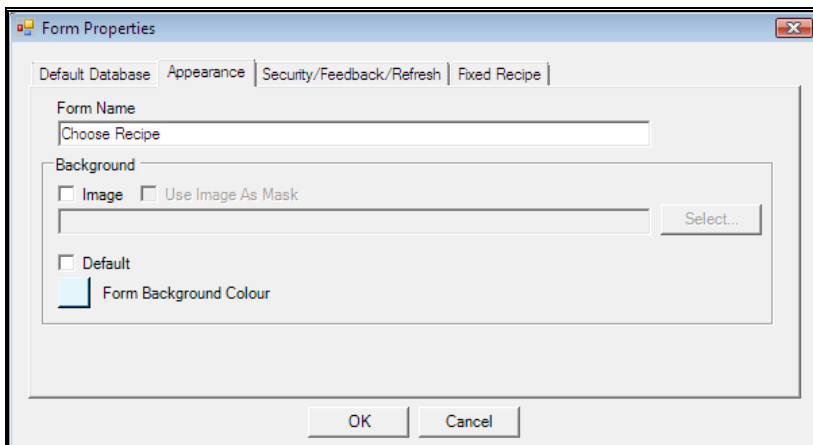
Click the **Select...** button next to the Database path to bring up the standard file chooser dialog. If the database you want to use does not exist, use the chooser to select the area and type the name of the database into the **File name** field.

If you select **Separate** then this means that the batches are created one per database, and the database file will have the same name as the batch. In this case only the path of the directory in which the databases are to be found needs to be specified.

- **Table.**

This allows you to set which table within the access database will be used as a standard by the form.

Appearance



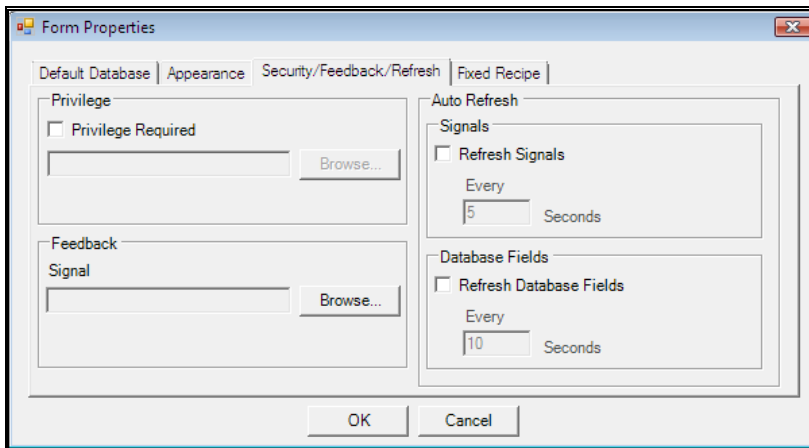
- **Form Name.**

This will appear as the title of the dialog when it is invoked (as Form Properties appears as the heading of the dialog above).

- **Background.**

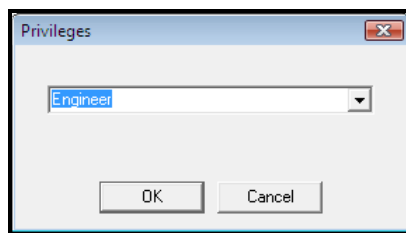
This allows the background to be set either as a image that can be chosen by marking the Image check box and pressing the select button or by picking a colour by pressing the square below the default check box. If the default check box is pressed the form will be the same grey that the above dialog box is.

Security/Feedback/Refresh



- **Privilege Required.**

This allows you to specify a privilege that the user attempting to run the XForm must have. This option is only necessary if you are implementing Prodigy Security. Select the privilege that is required in order to be able to run this form by using the Browse... button. This will display a list of all of the privileges currently available. If the Privilege Required field is left blank then any user will be able to run this XForm. The privileges selection box can be seen below. See the section on Security for more information on Privileges.

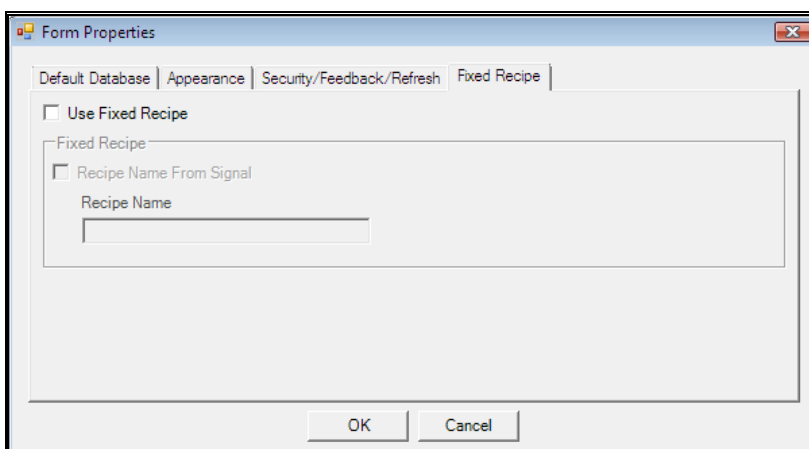


- **Feedback.**

This field defines the default signal that will be used in the feedback fields of button objects. If it is not entered here a signal can still be picked for feedback on a button by button basis.
- **Auto Refresh.**

This determines how often signals and database fields will be updated on the XForm; that is if they need to be updated at all. It can be useful if you are waiting for some variable to reach a certain value before you run a form.

Fixed Recipe

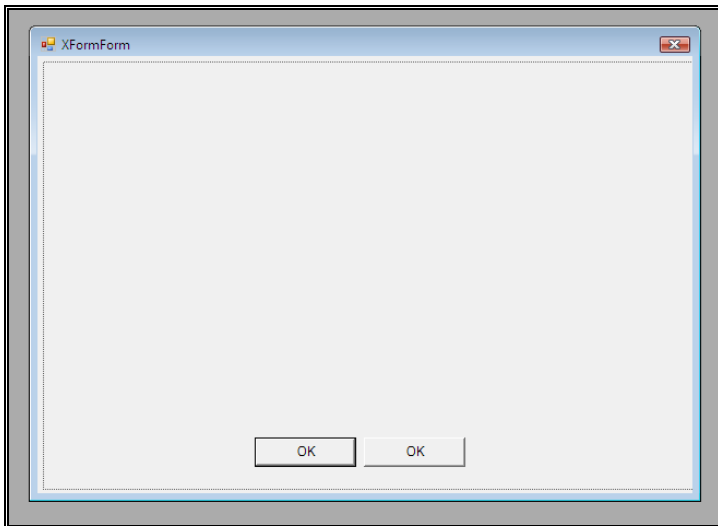


- **Fixed Recipe Name.**

Sometimes there is only one Recipe used by a process. If this is the case, there is little point in the

operator having to choose the recipe each time that the process begins. By specifying a fixed recipe name that recipe will automatically be opened when the form is invoked. If the **From Signal** box is checked and a valid signal is selected, the XForm will pull the recipe name from that signal. Otherwise the recipe name can just be specified in the Recipe Name edit box.

After setting up the properties for the XForm and clicking OK, you will be left with XForm that can be seen below. This is the XForm that has been created and the first thing to do is to drag it to an area roughly the size of the dialog you wish to create.



Adding Objects

There are several different types of objects that can be added to XForms, the buttons for these can be seen below and are explained in more detail later on. To add an object you must click on the appropriate button and then place the object on the XForm where it is required.



Pointer.



This is the pointer, it's used to select objects and perform such tasks as positioning and resizing or gaining access to the objects properties.

Label.



This allows text to be placed on XForms, the value of the text can be fixed or it can come from a signal or database field.

Edit Box.



This object allows signal or database values to be read into the XForm and viewed, it also allows data values to be entered by the user and written to a signal or a field in a database. Values can be read from a signal and written to a database or read from a database and written to a signal.

Check Box.



This can be used to set a digital signal to either a 1 or 0, it can take data from digital signals/ database fields in the same way that an edit box can..

Combo Box.



This provides a drop down menu where Signal/Database Field values can be selected from a list, the values can come from a finite state signal or the list can be created manually within XForm (described later).

Button.



This allows buttons to be added to the XForm; they provide a variety of functions such as open, cancel, save etc. they will be described in more detail later.

Grid Control.

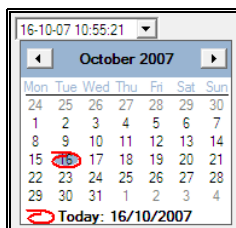


Data Grids provide a spreadsheet style of view onto a set of data that has multiple repeated rows, all keyed by a common one-off header record. One example would be Batch Data where a series of data items is logged repeatedly at regular intervals throughout the batch. These have very different properties to all the other objects that are available on XForms.

Date/Time.



This allows a date to be selected within a form; it is done using the date selector that can be seen below. The date can be read from a Signal/Database Field (must be in standard Prodigy time format) and written to another Signal/Database Field (Same as an Edit Box). There must be a destination for the data otherwise the field as a standard will be read only.



Picture Box.



This allows an image to be added to an XForm, the images location can be specified in a signal or it can be a fixed image specified at the time of design.

Radio Button.



Radio buttons allow signals to be set a specific value depending on which radio button the user checks; for them to function correctly a group of them must always be placed within a group box so that when you select one radio button the previously selected button becomes de-selected.

Group Box.

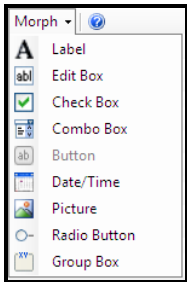


This allow a group box to be placed around similar objects to make the XForm easier to read and understand, it is essential that group boxes are placed around radio buttons for the radio buttons to function correctly.

Morph.



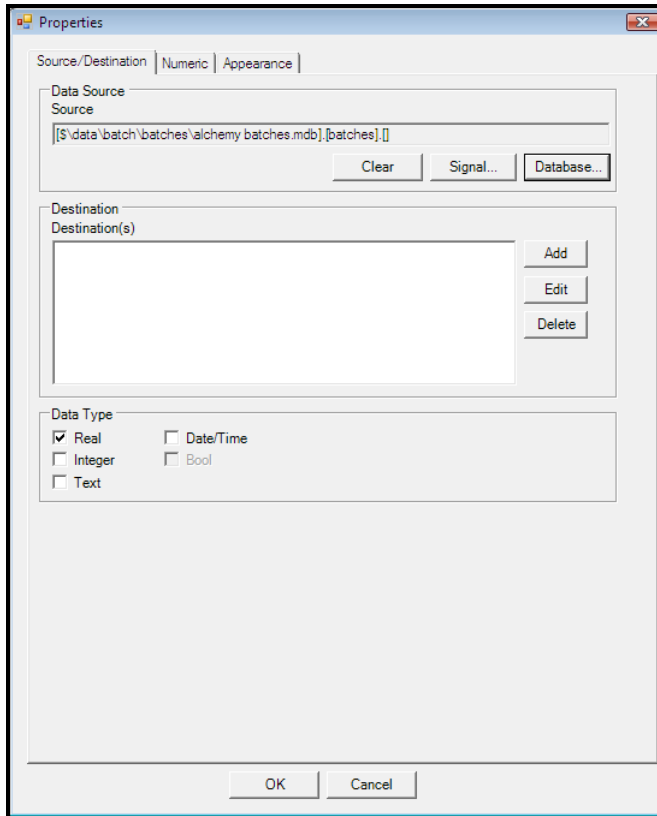
This allows one object on an XForm to be “morphed” into another e.g. and Edit Box into a Date/Time boxes. You must be careful when changing objects as the signals within the object will be transefered to the new object so it is unadvisable to change digital type objects such as check boxes into Date/Time boxes that require real numbers. The drop down menu can be seen below.



Properties of XForm Objects

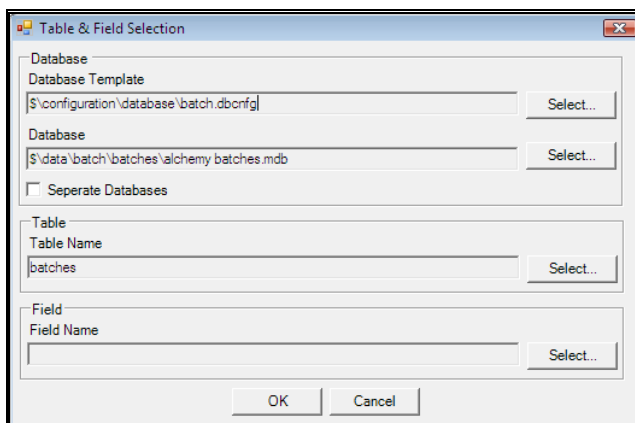
When selecting object properties it must be noted that many objects have similar properties, these being particularly the source/destination of data for the objects and the appearance of the objects. These communal properties will be described first and then the specifics for each particular object will be explained in more detail later.

Source/Destination



The dialog shown above shows the Source/Destination tab of an Edit Box. This is present on most objects though how the source/destination is used varies between objects (this is described in more detail later).

The data source can come from a Signal or a Database field; if it comes from a signal an appropriate signal must be selected using the SigScan dialog. If Database is selected the following **Table & Field Selection** dialog is displayed and is pre-loaded with default information that was specified in the XForm Properties (described earlier). The user then just has to select the required field.

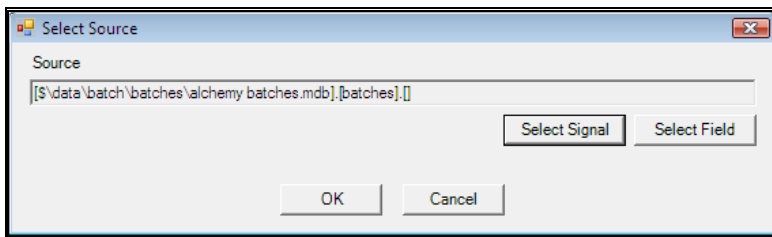


If values from a different database are required the appropriate Database Template, Database, Table and Field need to be selected from the required database.

Note that all databases used as the source and/or destination of field values must have been opened in some way by the XForm, most usually by a button object which has the Open property.



An object can have many destinations. Successive destinations are added or removed by pressing the Add or Delete buttons respectively. Pressing the Add button brings up the following dialog (pressing the edit button brings up the same dialog with the existing information relating to the selected signal loaded).



If the Select Signal button is pressed the SigScan Dialog appears, if the select field button is pressed the **Table & Field Selection** dialog is shown (described earlier). All destinations that a field is given will be sent the same value when an XForm is processed (i.e. when an OK or Apply action is performed), and will either be the value loaded from the source or a modified value entered by the user.

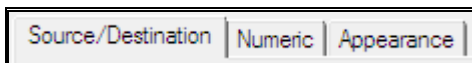
Note that destinations only apply to those object types that allow user input.

Some objects allow you to select the data type that that object is to display, while others have a fixed data type (e.g. a Check Box does not allow Text or Date/Time data and a Date/Time picker does not allow Integer, Text or Bool data). The types of data that are allowed for a particular object are shown at the bottom of the **Source/Destination** tab.

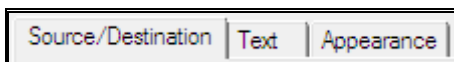


From the types that are available select the type that is appropriate to the signal or database field and how you want to use it. Depending on which data type is chosen the middle tab in the dialog will change, this can be seen below.

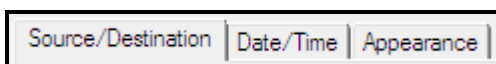
Real or Integer



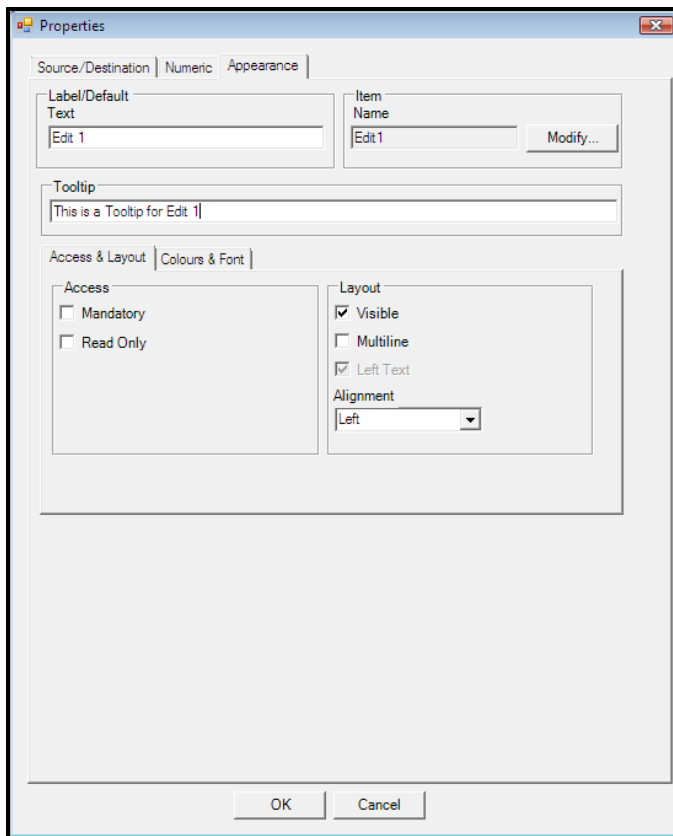
Text



Date/Time



Appearance

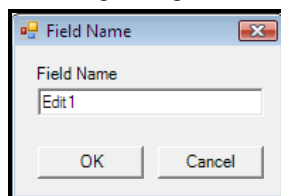


The dialog that can be seen above shows the Appearance tab of an Edit Box. The Properties are as follows:

- **Label/Default.**

This option applies in two instances.
The first is when an object has a label attached to it (e.g. a Radio Button or Check Box); in this case the value entered here will be the value shown in the label.
The second is if an object has a data type of Text or Numeric and has no source associated with it. The value entered here will be the value that will be displayed on screen when the XForm is run (in effect it is the default value).
- **Item Name.**

This is the name that is given to the object – in this case Edit1. If a data source is chosen for the object at the time that it is created, the field will be given the same name as the data source field as a default. This name will be used on any error dialogs that are displayed when the XForm is processed, e.g. if a mandatory field is left blank or a numeric field exceeds its upper or lower limits. Because of this it is sensible to give objects unique and meaningful name that will make them recognisable. To change the name of an object click the **Modify** button, which displays the following dialog.



- **ToolTip.**

There should be some information here describing the objects facilities; this text will be displayed as a ToolTip when the cursor moves over the objects position.
- **Access and Layout**

Mandatory: This is only necessary if the object can be edited; before an XForm is processed a check is made to ensure that there is a value within this object. If not an error is flagged and a

message is sent to the user.

Read Only: This option again only applies to objects that can be edited; it makes them read only so that the user cannot modify them. The user can then only see what will be going into the appropriate signal/database from the source.

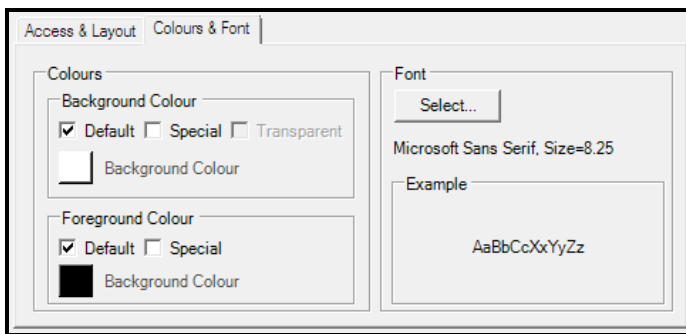
Visible: If this option isn't ticked the data will not be visible to the user. It will still be read from a source and written to a destination if this applies but the process will not be seen by the user.

Multiline: This option only applies to Edit Boxes; if this option isn't clicked the edit boxes are a fixed height. If it is clicked the edit boxes may be dragged to any height required.

Left Text: This option only applies to Check Boxes and Radio Buttons; as a default the Label/Default Text will appear to the right of Radio Buttons or Check Boxes. However if the button is pressed the text will appear on the left of the appropriate object.

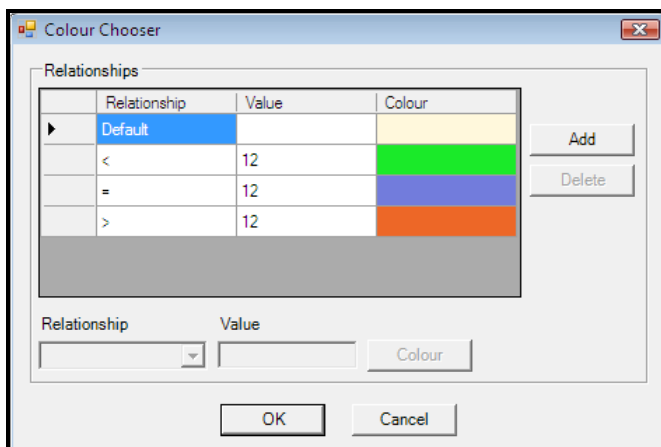
Alignment: This option only applies to Edit Boxes; it is the horizontal alignment of the text – left, Right or Centre.

- **Colours and Fonts**



Background Colour.

If **Default** is checked the colour is the default windows colour for the particular object type. If you uncheck default you can select a colour by clicking on the square next to background colour. Selecting **Transparent** makes the background colour of the object transparent so it will always be the same colour as the form background. **Special** allows colours to be associated with data values, the value that comes from the source of the object. Selecting a special colour will bring up the following dialog.



The relationship and value determine which colour will be selected for the object background. In the above example if the value of the source signal is less than 12 the background will be green, if it is equal to 12 the background will purple and greater than 12 will lead to an orange background. If none of the rules you specify lead to a colour being chosen (e.g. suppose you specified only < 12 and > 12 and the value is currently 12) then the Default colour is used.

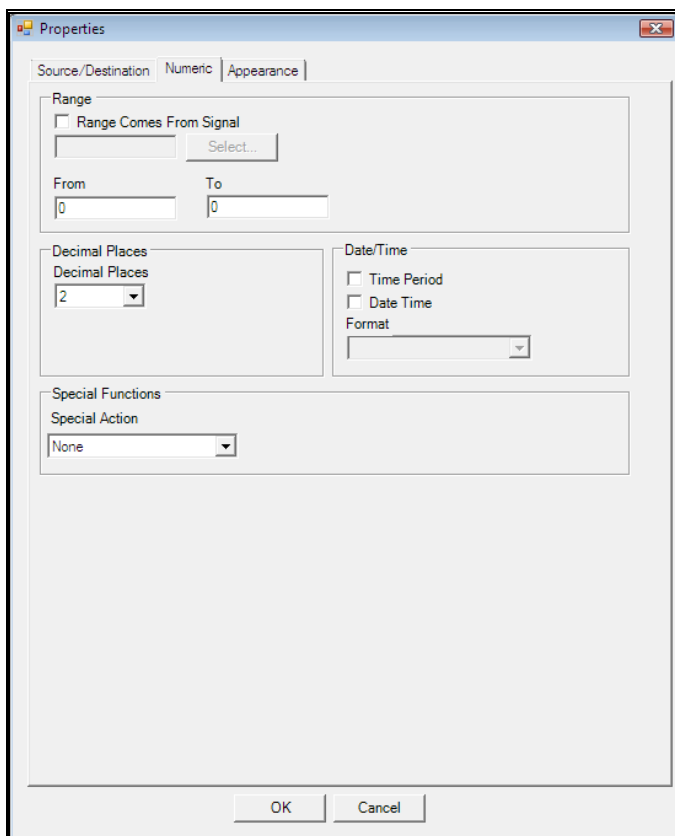
Foreground Colour.

This is the colour of the text in an object, the options are the same as for the background colour (excluding transparent).

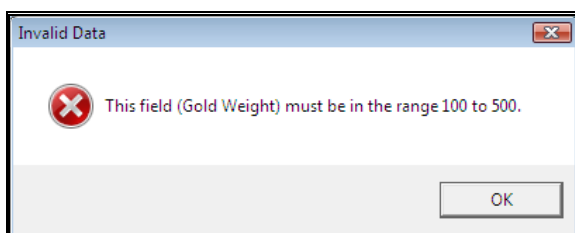
Font.

Pressing the select button on the font section allows you to change the properties of any text that can be seen; it brings up a standard dialog that allows aspects to be changed. What entered text will look like can be previewed in the “Example” area.

Numeric



- **Range.**
Validation can be performed on values that are entered into an XForms; you can either take it from a range specified in a signal by checking the **Range Comes From Signal** box and selecting a signal using the SigScan dialog or you can manually entered a range in the **From** and **To** boxes. If a range is used and you enter a value outside of that range and then try to process the XForm a dialog like the one below will appear and you will have to edit the required value before the XForm can be processed.



- **Decimal places.**
This is the number of decimal places that the value that is loaded will have displayed.

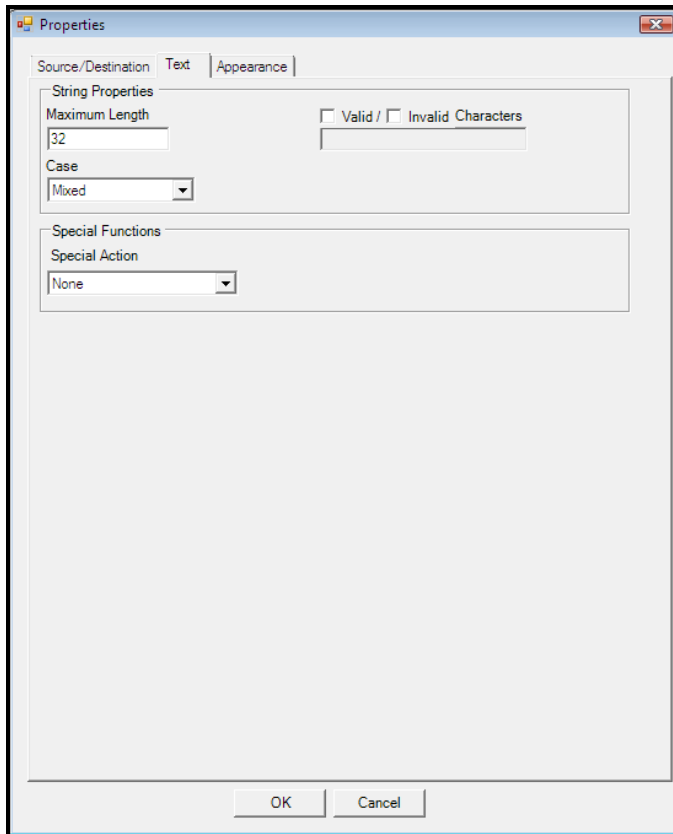
- **Date/Time.**

Numeric values can be formatted into dates or time periods, this can be useful if there is a large value for batch time and you want to see how many hours or days it's been running for. The format drop down menu is only available if one of the check boxes is selected and a range of different formats to suit specific applications are available, the type of format is different depending on whether time or date is selected.

- **Special Action.**

This can be used to get different signal parameters into or out of an XForm, such as the Alarm Limits or signal ranges.

Text



- **Maximum Length.**

This determines the maximum number of characters that can be written into an edit object that can use text.

- **Case.**

Mixed case.

The text is not altered in any way.

Upper case.

The text is converted to upper case before being passed on to its destination.

Lower case.

The text is converted to lower case before being passed on to its destination.

Leading Sentence.

The text is converted so that the first letter of the string is a capital and the rest is lower case.

Leading Word.

The text is converted so that the first letter of every word is a capital and the rest of the letters in each word are lower case.

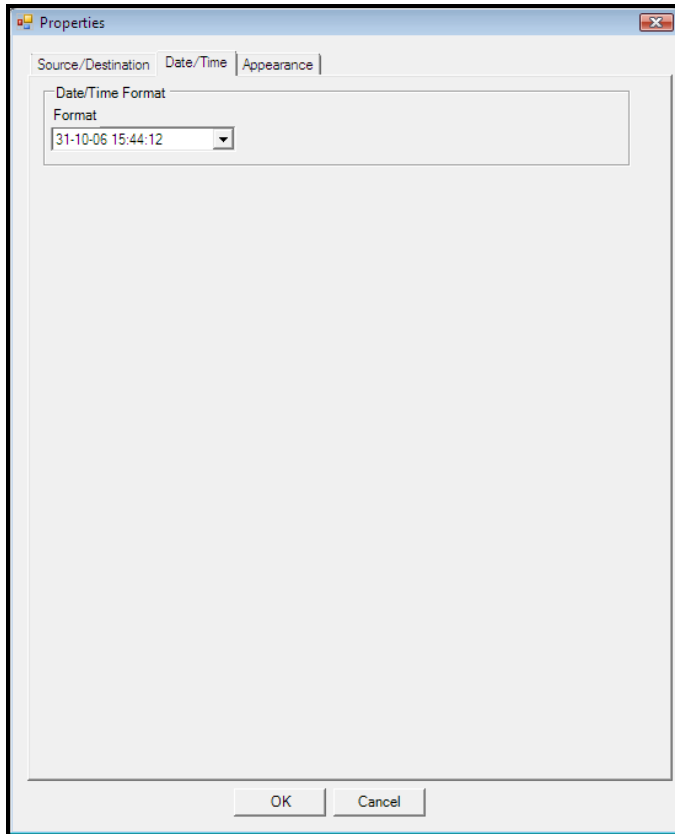
- **Valid/Invalid Characters.**

Characters specified here will be available for entry if valid is selected or unavailable for entry if invalid is selected.

- **Special Action.**

This can be used to get text values relating to signals (e.g. alarm texts, descriptions etc.) or to load dialogs that can get special values in (e.g. user names, file paths etc.).

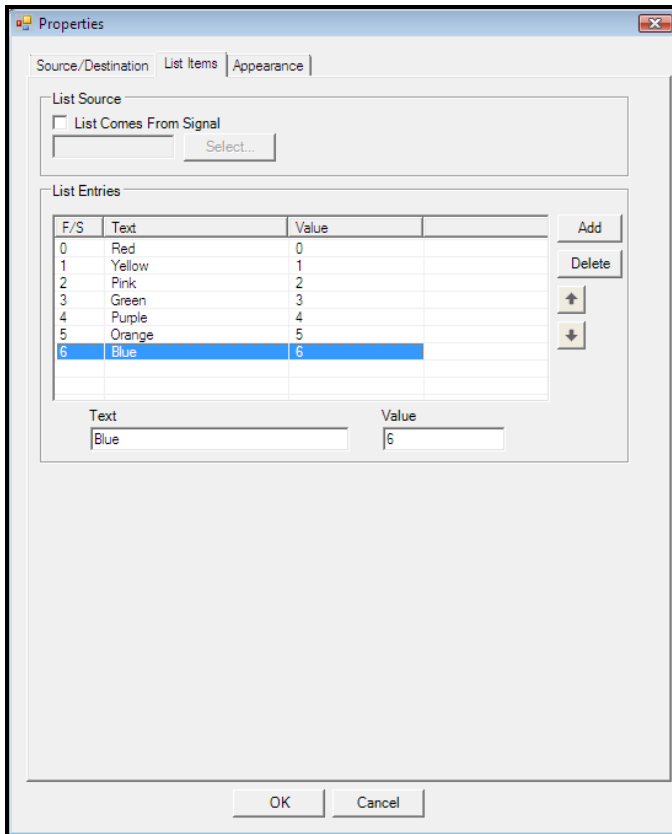
Date/Time



- **Date/Time Format.**

Allows the user to select the format of the date that they want to be displayed on an XForm.

List Items



- **List Source.**
List items can either come from a signal (Digital or Finite State) or a list can be created by the user. To get the values from a signal simply check the **List Comes From Signal** box and select the appropriate signal from the sigscan dialog.
- **List Entries.**
To create a list manually click the **Add** button; enter the text you wish to be displayed in the drop down list and the value that you want to be sent to the signal if that item is selected. You can delete list entries by pressing the **Delete** button or change the order by selecting a list entry and using the up or down arrows.

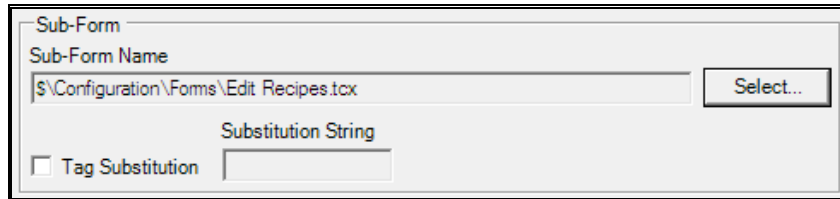
Buttons

Type	Database	Table
Primary	\$\data\batch\batches\alchemy	batches.mdb batches

- **Button Action.**
Defines what action is performed when the button is pressed.
 - **OK.**
Pressing the button causes the Apply action (see below) to be run and on successful completion of that action the form is dismissed.
 - **Apply.**
Pressing the button causes data to be written to the batch and recipe databases and to any signals, as specified by the individual fields on the form. Data validation is performed on the fields - mandatory fields may not be left blank, numeric fields must contain numbers, etc. If validation fails a warning message is issued and the cursor is placed in the offending field. No data is written away until all fields are validated successfully.
 - **Cancel.**
Pressing the button dismisses the form but does not cause any data to be written to either the batch or recipe database, or to any signals.
 - **Delete.**
Pressing the button will delete the currently selected batch and/or recipe. This button action will only be used on an XForm designed for "housekeeping" of the relevant databases. A warning is issued before any deletions are actually performed, to ensure the button is not pressed accidentally.
 - **Open.**
Pressing this button will cause the Batch Selector dialog to be displayed. This will open the batch database specified in the XForm properties and display a list of all of the batches currently in the database.
When the Open action is specified for a button the **Open** section of the above dialog becomes available and clicking the edit button will bring up the *Primary Database* dialog that can be seen at the end of this section.
 - **Refresh.**
Pressing the button causes all fields on the XForm to be re-read from the relevant sources specified (batch or recipe file or signal database).

- **Sub-Form.**

If this option is selected then a second *child* XForm is run. It will use the same batch and/or recipe database as has already been opened in the parent XForm. Use this option if an XForm would be too large and unwieldy to implement as one. When this option is selected the **Sub-Form** section becomes available, this can be seen below.



The form that you want to open can be selected using the **Select** button near the **Sub-Form Name** edit box; tag substitution can also be performed by selecting the relevant check box and entering a Substitution String.

- **New Form.**

If this option is selected then a second XForm is run. The XForm that is to be run will need to be specified in the sub form section as above however it will inherit **no** information from the XForm that it has been run from.

- **Save As.**

If this option is selected, the current record that has been opened can be saved to the database with a new batch name that is entered by the user.

- **Button Label.**

This is the text that will appear on the button. Defaults are chosen as appropriate to the button action selected, but you are at liberty to change this as required.

- **Write This Field Last.**

Only applicable to OK and Apply actions, this allows you to specify the name of the field that is to be written last. An example of why to use this is that perhaps a SLANG routine is waiting for a certain value to be set before performing some actions. If those actions are performed on other values that are set by the XForm then it is imperative that this trigger signal is the last one to be set, otherwise the SLANG routine may perform actions using values that are *about* to be changed as successive values on the form are written away.

- **Feedback**

The feedback section of the form allows a signal to be specified that will be set to a certain value when the button is pressed. This is useful for setting a signal to start or stop a batch, or to let SLANG know what the XForm has done. For example an OK button may set a signal to the value 1 but a Cancel button might set it to 0. This will allow a SLANG program to know how the XForm was exited and allow it to take appropriate action, for instance secondary validation of the saved data if OK was pressed.

The signal isn't set until any validation that is required has been successfully completed on all of the values within the form.

Primary Database

Note that as well as the Primary Database a **Secondary Database** must be configured if **Data Grids** are going to be used on the form. Standard XForms access header data (one line per batch) while Data Grids access repeating lines of data (multiple lines per batch) held in separate tables (In a standard Batch database there are tables for continuous data, alarms and comments, all of which allow multiple lines per batch). Setting up the open function for a secondary database is detailed in the section on how to configure a Data Grid.

Field Name	Display As	Sort	Desc.	Filter String
Batch Name	Batch Name	Yes		
Batch Start	Batch Start			
Batch End	Batch End			

- **Database.**

This is where the information on the database that is to be opened needs to be entered; the default values loaded will be those in the default form properties.

- **Table.**

This is the name of the table in the database where data will be taken from.

- **Fields.**

These are the fields from the selected table that will be displayed in the batch chooser to allow the user to identify the appropriate batch/recipe that they want to select.

- **Display As.**

This is the title that the user will see on the batch chooser above each of the fields that have been added. Data that is shown in the batch chooser can be sorted; to do this check the **Sort** box, as a default records will be sorted in an ascending order but this can be reversed by checking the **Descending** box.

- **Selected Record Must Be.**

If the check box that is labelled *Existing* is selected, an existing batch name must be selected from the batch chooser. If the check box *New* is selected, a new batch name must be entered. If neither or both are checked an old batch may be selected *or* a new batch created.

- **Filter String.**

A filter can be applied to a field and the string that is entered here will determine which records are available for selection.

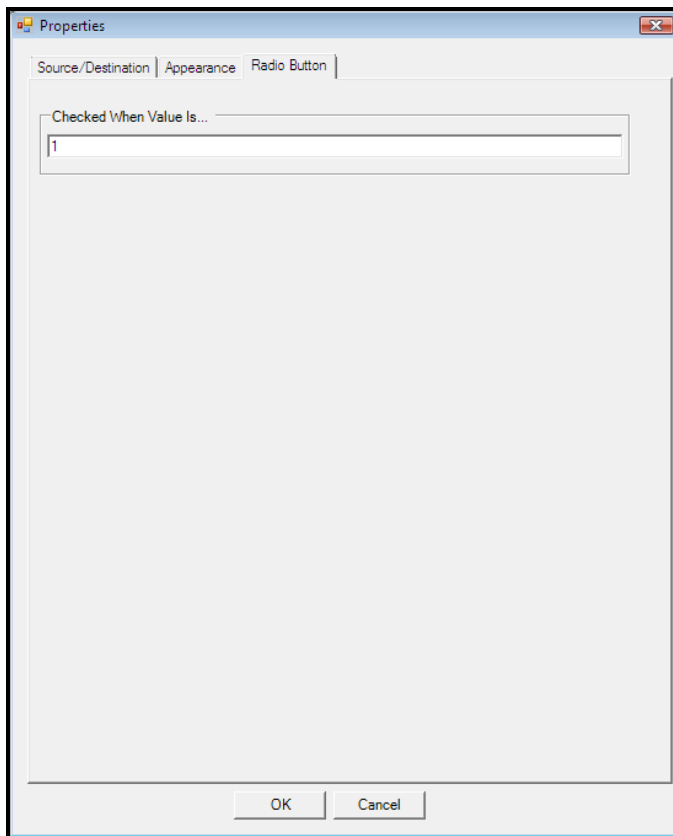
If the **Display “Show All” Button** check box is selected there will be a button on the batch chooser that allows the filter string to be overridden and all records displayed.

If the **Allow “On The Fly” Filter Changes** check box is selected this will allow the filter string to be modified to show different records.

Check Boxes

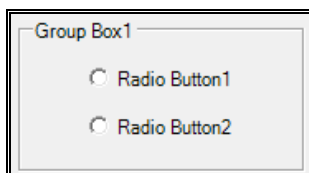
Check boxes only have the source/destination and appearance tabs on their properties dialog. In general they require boolean signals however they can have integer or real signals associated with them. In relation to the source the check box will appear empty if its value is zero and checked if its value is 1 or not zero (in the case of integer and real numbers). When writing to destinations a value of 1 or zero is written to the destination depending on whether the check box is ticked or not.

Radio Button

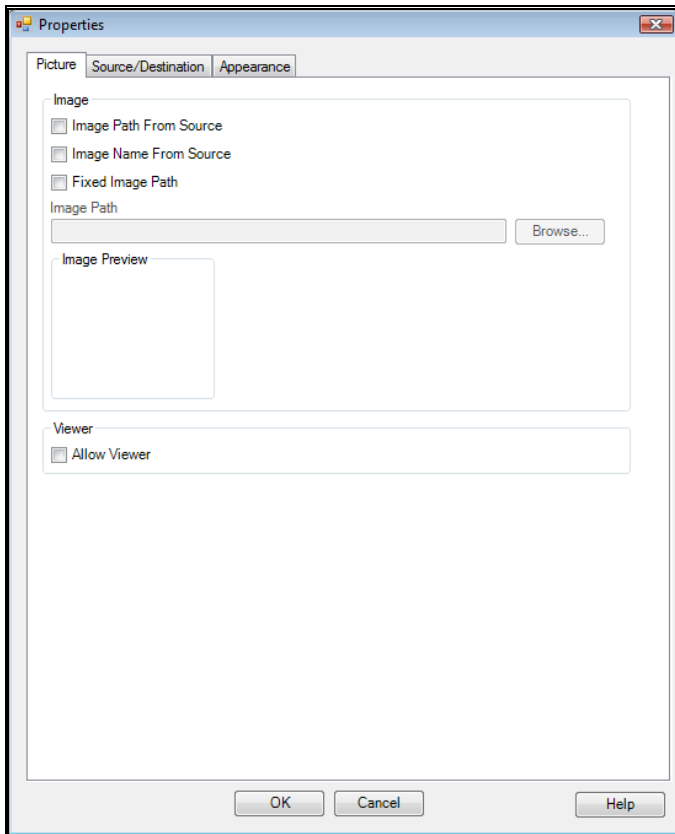


- **Checked When Value Is...**
This is the value that will be written to the destinations when a particular radio button is checked.

In order for radio buttons to work they need to be surrounded by a group box so that the XForm can decide which radio button needs de-selecting when a new one is selected, an example of this can be seen below.



Picture



- **Image.**
Defines where the image displayed in the object comes from.
- **Image Path From Source.**
If this option is selected then the path of an image that is to be displayed on the XForm can be specified in a signal or database field that is selected using the Source/Destination tab.
- **Image Name From Source.**
If this option is selected then the name of an image that is to be displayed on the XForm can be specified in a signal or database field that is selected using the Source/Destination tab. The folder in which the image is stored needs to be specified using the **Browse** button next to the Image Path section.
- **Fixed Image Path.**
If this option is selected then the file location of a fixed image can be selected and this image will always be shown on the XForm.
- **Viewer.**
If **Allow Viewer** is checked the image that is selected will still be displayed on the XForm at the specified size however if the user clicks on the image a full size viewing window will be loaded that will allow the user to see the image at its full resolution.

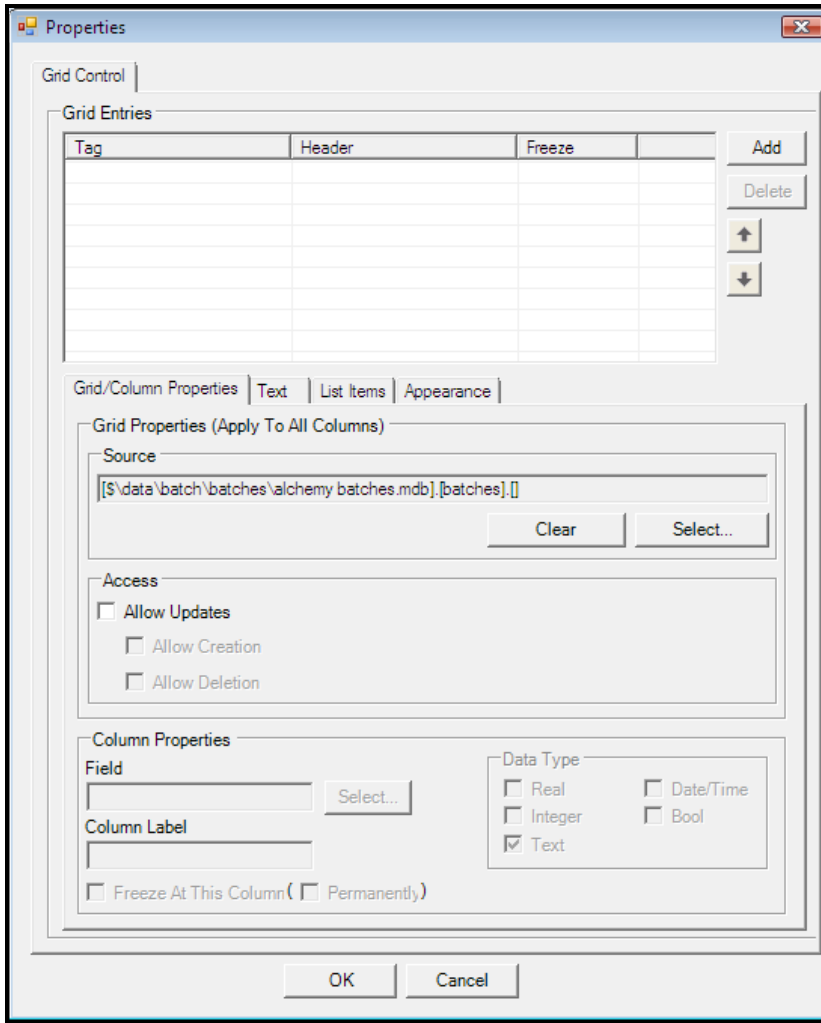
Data Grids

Data Grids provide a spreadsheet style of view onto a set of data that has multiple repeated rows, all keyed by a common one-off header record. One example would be Batch Data where a series of data items is logged repeatedly at regular intervals throughout the batch. If you use Continuous Recording in **BDR**, this will produce a table of this sort.

To add a data grid click the Data Grid button on the XForm Items Toolbar.

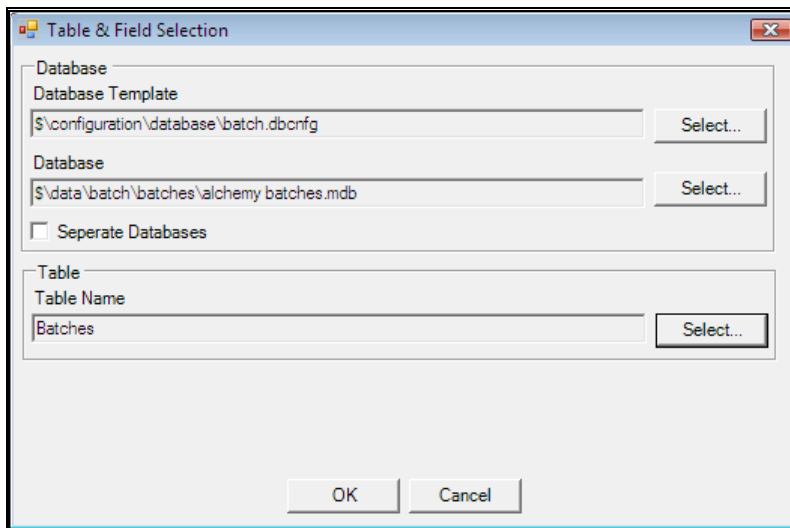


Drag an area on the XForm. Right-click on the data grid to bring up its properties dialog.



Grid/Column Properties

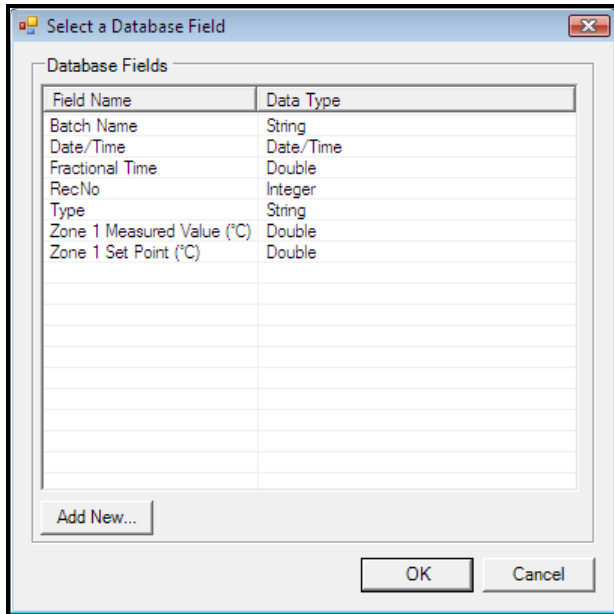
- Source.**
 The Access Database Configuration file (Database Template), the Database and the table that will be used in the Data Grid need to be set by pressing the **Select** button. This is done using the dialog that can be seen below.



The **Table** that holds the repeated information needs to be selected or the Data Grid will not work. For the standard **Batch** database, this table is the **Details** table. Note the XForms Default Database information is loaded into the Table and Field selection dialog as standard.

- **Adding Data Grid Fields.**

This is done by pressing the **Add** button; pressing it will bring up the dialog seen below that allows selection of the fields that are to be displayed within the Data Grid. Select and add a required field and repeat the process until all the fields are present that you want displaying in the Data Grid.



When all fields have been selected you can delete a field by pressing the **delete** button or change a fields position by selecting it and pressing the **up** or **down** arrows. The order of the list of fields is the order that columns appear on the grid.

- **Access.**

If **Allow Updates** is checked then information within the data grid can be edited; when it is checked the **Allow Creation** and **Allow Deletion** check boxes also become available. **Allow Creation** lets the user add new records into the selected data set and **Allow Deletion** lets the user delete records from the data set, this can be useful for deleting erroneous data points from a data set.

It must be noted that you cannot limit which columns the user has access to; either all columns can be edited or none, therefore this must be taken into consideration when picking which fields to display in a Data Grid.

- **Column Properties.**

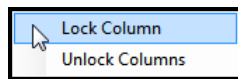
When a field has been added and then selected the appropriate field name is displayed in the Field edit box. The Column Label by default will be the same as the field name but this can be changed by editing the value in the **Column Label** edit box.

The **Freeze At This Column** check box freezes the position of the selected column and all columns to the left; it will hold them at the left of the screen if the scroll bar is used to view other columns. This can be seen in the two images below where **batch name** is held at the left even when the user has scrolled fully to the right (the second image).

Open		Batch Name 20070820TEST			
	Batch Name	Date/Time	Type	Zone 1 Measured Value (°C)	Zone Point
▶	20070820TEST	20-08-07 16:36:21	0	10.00	0.00
	20070820TEST	20-08-07 16:36:23	0	10.00	0.00
	20070820TEST	20-08-07 16:36:25	0	10.00	0.00
	20070820TEST	20-08-07 16:36:27	0	10.00	0.00
	20070820TEST	20-08-07 16:36:29	0	10.00	0.00
	20070820TEST	20-08-07 16:36:31	0	10.00	0.00
	20070820TEST	20-08-07 16:36:33	0	10.00	0.00
	20070820TEST	20-08-07 16:36:35	0	10.00	0.00

Open		Batch Name 20070820TEST			
	Batch Name	Zone 1 Set Point (°C)	Fractional Time	RecNo	
▶	20070820TEST	0.00	0.45	4048	
	20070820TEST	0.00	0.48	4049	
	20070820TEST	0.00	0.52	4050	
	20070820TEST	0.00	0.55	4051	
	20070820TEST	0.00	0.58	4052	
	20070820TEST	0.00	0.61	4053	
	20070820TEST	0.00	0.64	4054	
	20070820TEST	0.00	0.47	4055	

When an XForm is running the user can right click on one of the column headings and manually lock or unlock columns.



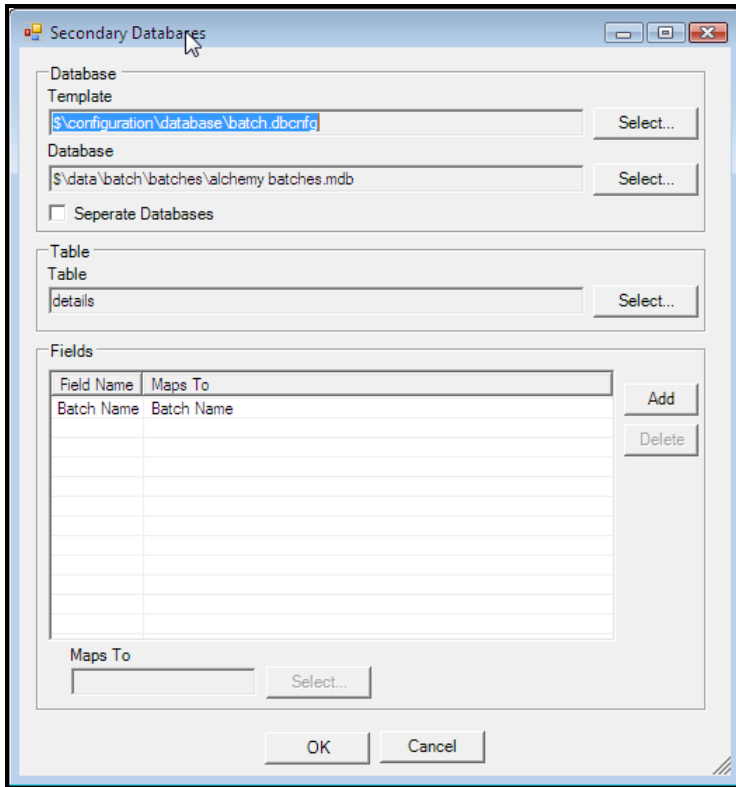
Selecting the **Permanently** check box prevents the user from being able to manually unlock columns when an XForm with a Data Grid is running.

The Data Type set of check boxes allows the user to select the relevant data type for the field. The data type that is selected will cause the tabs to change in a way similar to how the tabs do when selecting a data type for other XForm objects.

The **Appearance** tab is always present as is a tab relating to the data type that has been selected all items on these tabs have been previously described in this section of the manual. If the data type that has been selected is anything other than date there will also be a **List Items** tab. This allows you to create either a list of text items or a list that can map a text value against numeric value. When the grid is displayed this field will appear as a drop down list of text items. In the case of mapping text values against numeric values selecting an item causes the value of the number that is assigned to this list value to be stored

Secondary Database

Information in a data grid needs to be loaded from a different then the other information in an XForm due to the different structure of the data (normally continuous data stored in the details table). An additional table needs to be opened to get the data and this is done by adding a secondary database to the button that performs the **Open** function. To do this click the **Add Secondary** button in the **Open** section of the Button tab which will bring up the following dialog. Information can then be entered about the data that you wish to load into the Data Grid.



The information that is required is the database template, database name and the table within the database. The first two of these fields should be the same as for the Primary Database but the Table field will be different. The Table field should refer to a table in the database that contains repeated lines of data all keyed by the same field(s) that form the Primary Key of the Primary Database. For example a standard Batch Database has Batch Name as the primary key of the Batches table and this field also keys repeated lines of data in the Details, Alarms and Commentary tables, all of which can be displayed in Data Grids.

The Fields section then needs to be configured to say which field(s) on the XForm (read from the primary database) provide the keying information required to select records from the secondary database. In most instances this will just be one field (in the Batches example mentioned earlier, it will be Batch Name). The required field(s) from the secondary database are selected and then “mapped” to an appropriate field on the XForm that will be read from the Primary database.

When the Open action is performed the primary database table is opened first and any values on the XForm are read in. Then any secondary database tables are opened using the specified value(s) just read from the primary database as keying values for the secondary table. All records matching that key are read in and their values displayed in the relevant Data Grid fields.

Show and Auto Label

Show and auto label only apply when running in design mode; show provides different views of objects and auto label allows control over the automatic labelling that takes place when creating certain objects.

Show

The default setting for this feature is **Source/Destination**, this can be seen below; items that have a source or destination display the appropriate signal or field in the object. If an item only has a destination (as with **/\$OPERATOR** below), the destination name appears after a forward slash (/) if there is only a source the source name appears before the forward slash. If a source and some destinations are present they are separated by the forward slash and multiple destinations are separated by commas after the forward slash.

Often several objects may need to be aligned in order to improve the neatness of the XForm or to ensure co-incident of positioning. Six buttons on the formatting bar provide alignment functions. Select all of the objects to be aligned, being sure that the last one selected is the one you want to align the others to, this being known as the dominant control (its selection outline and handles are a different colour than the other objects, to highlight its dominance). Press the relevant alignment button, these being, from left to right

- Line up left edges
- Line up right edges
- Line up top edges
- Line up bottom edges
- Centre horizontally (i.e. horizontal position is affected).
- Centre vertically (i.e. vertical position is affected).

Sizing Objects



Sometimes you may want several objects to be the same size as each other. You can do this by using the three formatting buttons shown above. Select the objects to be sized, ensuring that the last one selected – the dominant control – is the one you want all the other objects to be the same size as. Then press the required button. The function of the buttons is, from left to right:

- Make objects same size
- Make objects same width
- Make objects same height

Spacing Objects



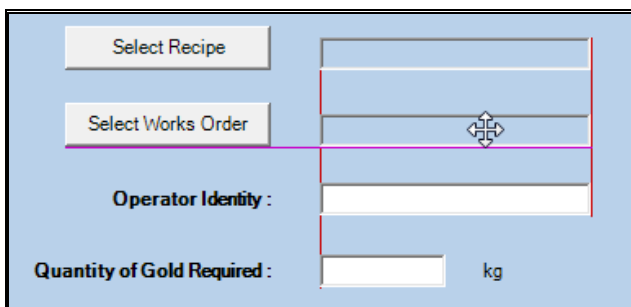
At times you may wish to space several objects evenly across an XForm. To do this, select all of the objects that you wish to space and use one of the two buttons on the formatting bar that are show above. The left button spaces objects across the screen, the right one spaces them down the screen.

To space objects across the screen, XForm takes the left and right extremities of a bounding box that would contain all of the objects. Any objects that are at these extremities remain unaffected and the other objects are positioned so that there is an equal sized gap between all of the objects being spaced.

Guide Lines and Magnetic Guide Lines



There are two types of guide line available for use; they can be used independently or together. The first is the standard guide lines, this places a guide line when two objects edges are exactly in line this affect can be seen in below.



If Magnetic Guide Lines are turned on; when an object is been moved and one of its edges becomes roughly in line with the edge of another object, the one you are moving will “snap” into alignment with the second

object. Small movement will not remove it from this position, the mouse cursor needs to be moved several pixels away before the object is free to move again. Both types of Guide Line can be used at the same time and they make aligning objects a lot easier.

Duplicating Objects



Select the object to be duplicated and then click the **Duplicate** button (above) on the formatting bar. The new object will overlay the existing object, offset by a few pixels. Drag it to its required position. All attributes, including the signals or database fields selected, are copied from the old object.

Set Tab Order



If this button is selected the tab order is turned on; this can be seen as numbers next to each of the objects on the screen that tabbing applies to, below is how the tabbing order may look on a form.

The screenshot shows a 'Choose Recipe' dialog box with the following elements and their tab order numbers:

- Select Recipe: 5
- Edit Recipe: 9
- Select Works Order: 6
- Operator Identity: 3
- Quantity of Gold Required: 4
- Colour: 20
- Raw Materials section:
 - Mercury: 10
 - Copper: 11
 - Lead: 12
 - Sulphur: 13
- Fusion Details table:

Stage	Temp	Time
1	14 °C	17
2	15 °C	18
3	16 °C	19
- Accept: 7
- Cancel: 8

To change the tabbing order just click on objects one at a time in the order you want to tab through the form; if you make a wrong selection you can start again by clicking on the **Set Tab Order** button twice to turn it off and then back on again.

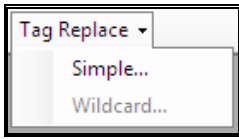
If you want to start changing the tab order from some point other than the first object <Shift> click the tab square that you want to start from and then click on the next object that you want to Tab to when using the form.

Detach Group Box



If this is selected any items that are within a group box are no longer attached to that group box; so if you move the group box then items within will not move with the group box like they normally do.

Tag Replace



If this is used tags can have the first characters replaced as can be done in Display Builder; **Simple** Tag Replacement can be done through the dialog that can be seen below where the user simply enters the characters that they wish to be replaced.



At present **Wildcard** tag substitution has not been implemented.

XForm Design Requirements

Selecting a Batch

- You must place a button on the XForm that has the Open Batch function. Decide whether the batch/recipe must be old, new (naming a new batch) or whether this doesn't matter.
- You must enter the location and name of the batch/recipe databases to be used in the XForm Properties dialog.

Showing the Selected Batch on a Display

- Create a string derived signal to hold the name of the current batch and make this the destination of a XForm edit box that contains the batch name for the current batch.
- Create a text Display Builder item whose Value Display property shows the text associated with this signal.
- The batch name will not appear on the Display until the **Apply** button on the XForm has been pressed and the value has been written to the appropriate signal by the XForm.

Starting a Batch

- Create a Digital derived signal that shows the state of the batch (e.g. 1 running, 0 stopped)
- Place a button on the XForm that has the **Apply** or **OK** action.
- In the feedback section of the button properties set the value of the signal to 1 hence starting a batch.

Ending a Batch

This is as for "**Starting A Batch**" but the signal should be set to 0 (assuming the start and end conditions in BDR are 1 and 0).

Recipes

Recipe information is handled in an identical manner to batch information. Make sure you do not use the same signal to control batch and recipe names and states. The only difference is that the start and end times for recipes are irrelevant.

Using an XForm

XForms can be launched in two way:

- Place a button object on a Display and give it the action "**Launch XForm**". When the Display is running and this button is pressed the XForm specified will be launched.
- Add an item to the **Menu Bar**. Specify XForm as the program name and the path of the form configuration (.tcx) file as the final parameter. This will also work for **Directory Based** menu items, simply specify .tcx as the extension.

Initially the fields on the XForm may be completely blank or have a value taken from specified sources. Generally, any fields that have a batch or recipe as their source will be read-only.

Select the recipe to be used (if any) by clicking the relevant button on the XForm and either choosing an existing recipe or entering the name of a new one.

Select the batch to be used (if any) in the same manner as for the recipe.

All of the fields whose source was a batch or recipe should now be editable, unless they were specified as being read only. It is usual that any fields that hold recipe information when starting a batch *will* be specified as being read-only, unless the operator is allowed to override recipe values.

*It is not possible to transfer information from a recipe to the signal database and/or batch database without it appearing on the XForm. Some fields may well therefore be simply **transit** fields, read-only fields used to pass the information from source to destination. These can be made so that they are not visible to the operator.*

Enter any data required into the fields and press the Apply (or similar) button. Validation will be performed on the data in the fields, checking it against any ranges specified and ensuring that all mandatory fields have been filled. In addition, the data type is checked for validity, so that a text string cannot be entered into a field expecting a numeric value.

Any errors are displayed via a dialog and the cursor is positioned in the offending field. Correct any errors and try again.

An Example XForm

Below is an example of a recipe selection/batch starting XForm from the Alchemy Demonstration.

Stage	Temp	Time
1	<input type="text"/> °C	<input type="text"/>
2	<input type="text"/> °C	<input type="text"/>
3	<input type="text"/> °C	<input type="text"/>

Recipe information is at the bottom of the screen and the fields are read-only as they are for information/transit purposes only. The information in the fields is written to the batch database and also into signals that control various aspects of the Alchemy production line.

Batch information is at the top of the screen and the operator must enter the **Works Order Number** (Batch name), **Operator Identity** and **Quantity of Gold Required** after they have selected a recipe.

When all of the information is entered the user would click the Accept button which causes it to be written to its destination(s).

XForms as Reports

XForms may be used to provide a basic batch reporting functionality. Create a form that selects a batch and displays the required information in read-only fields. Information that can be displayed includes the batch name, start and end times, recipe name, recipe values, etc. In fact, any information written to the batch file by any other XForms and/or **BDR** can be displayed (provided it was for the same batch name).

In the XForm below fields show the batch name, start and end dates, gold rating and a grid shows some continuous batch data recorded by BDR. A sub-form launched by the Other Details button additionally shows the cost of making the gold and the profit we have made.

Date/Time	Log Type	Zone 1 Measured Value (°C)	Zone 1 Set Point (°C)
22-08-07 08:55:54	Automatic	10.0	
22-08-07 08:55:56	Automatic	10.0	
22-08-07 08:55:58	Automatic	10.0	
22-08-07 08:56:00	Automatic	10.0	
22-08-07 08:56:02	Automatic	10.0	
22-08-07 08:56:04	Automatic	10.0	

Make sure that the only buttons on an XForm Report are to choose the batch, show other details (launch a sub-form) and to exit the form. Make sure no buttons perform any data writes.

The Batch and Recipe Databases

- Batch/recipe databases are held in Microsoft Access™ or SQL format.
- The creation of the databases and all records and fields within them are handled by Prodigy. Do not tamper with the databases by using Access or another third party database program unless you know what you are doing.
- There is no limit to the number of different databases that can be created. You may want to have a different batch and recipe database per plant area.
- Within each database there is no effective limit to the number of records that can be created (there is one record per batch or recipe).
- Within each record there is a limit of 255 fields per record when using Access™ but when using SQL there is no exact defined value as it is dependent on the total number of objects in the database. All records within a given database have the same fields.

Batch Chooser Configuration

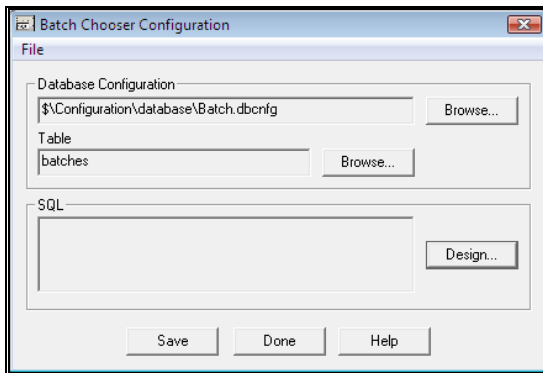
The Batch Chooser

The Batch Chooser is the dialog that appears when you need to select a batch. In its standard form it shows all of the batches in a batch database in a list containing three columns:

- Batch Name
- Start Date
- End Date

You can design your own Chooser Configuration to be used by Historical Trends. This allows you to allocate a column in the Batch Chooser dialog for any field that appears in the batch database, and also to filter the records by supplying an SQL string that will be applied to the batch database.

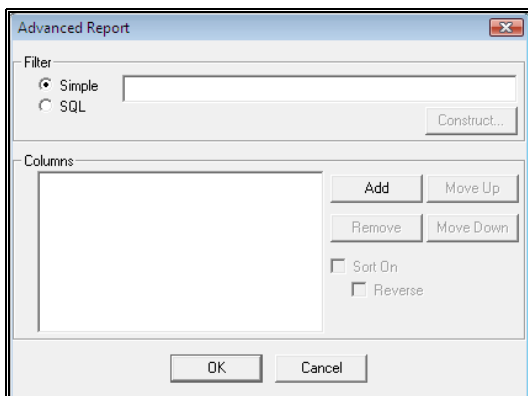
The main dialog of the Batch Chooser Configuration program is shown below.



To open an existing configuration click on **File** and then **Open...** To start a new configuration when an existing configuration is already loaded, choose **File** and then **New**. To save a configuration either click **File** and **Save** or **File** and **Save As...**

To modify or create the configuration click **Browse...** to select the database configuration file to be used by this configuration. When you use this batch chooser configuration it can be applied to any database so long as the fields used within this configuration (see later) exist in the database too.

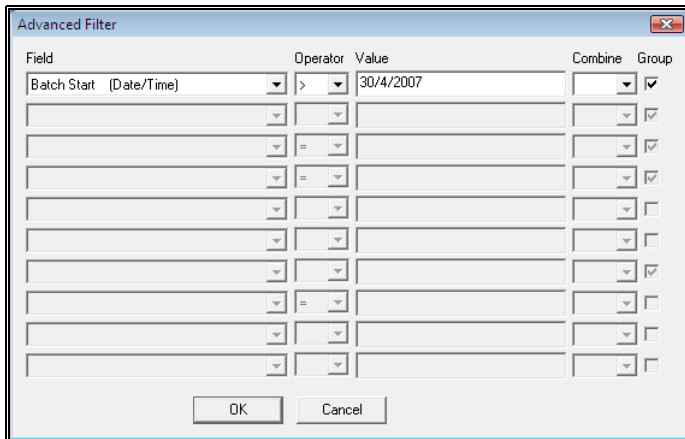
To design the actual Batch Chooser, click on the **Design...** button to display the dialog shown below.



The upper part of the dialog is the **Filter**. This can be either **Simple** or **SQL**.

If the filter type is **Simple** then you provide a simple comparison string for the Batch Name field in the database. For instance, if the string is "ABC", the Batch Chooser will show all batches whose Batch Name contains the string ABC. You do not need to select a database on the main dialog to use this option.

More sophisticated filtering can be achieved with the SQL filter. For this, a database must have been selected on the main dialog. You can either type the SQL string in directly, or you can click the **Construct...** button to display the dialog below, which helps you to build an SQL statement.

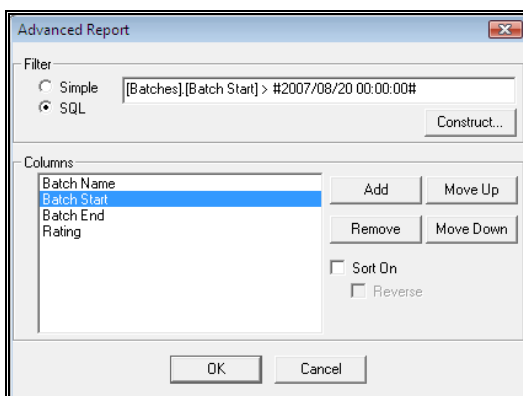


To build an SQL statement perform the following actions:

- Select the field that you want to filter on. The drop-down lists at the left of the dialog contain all of the fields in the database, along with an indication as to their type.
- Select the required logical operator from the next drop down list.
- Enter the value that you want to compare the field against in the **Value** box. There is no need to put “ marks around strings, or to put # marks around dates. The dialog will do this automatically when it exits.
- If there is another comparison you wish to make, select **AND** or **OR** from the drop down list under **Combine**.
- Complete any subsequent rows,
- You can group statements together (this has the effect of “bracketing” the statements. Check the **Group** box at the right of a row to begin one group. Check a second row to mark the end of the first group and the start of the second. You do not need to mark the final entry, a group will automatically end at the last row of comparisons.

When you exit this dialog with **OK** it converts the selections you have made into an SQL string that is displayed in the **SQL** box of the main dialog.

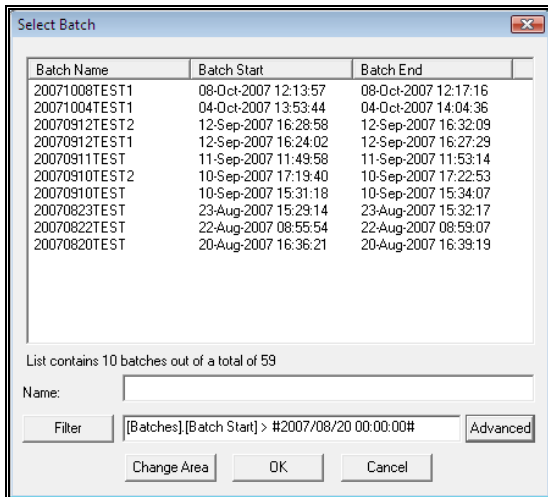
The **Columns** group at the bottom of the **Advanced Report** dialog is for defining which fields are to be displayed as columns in the Batch Chooser dialog.



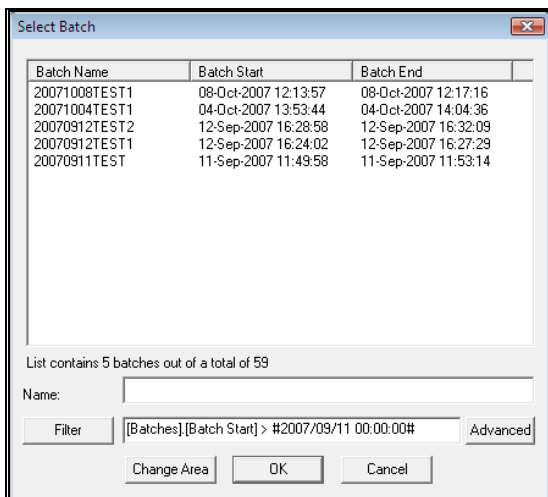
- **Add.**
Adds extra fields into the list.
- **Move Up/Move Down.**
Arranges the order of the fields. Note that the Batch Name must always be in the list somewhere. When the Batch Chooser is displayed, the columns will appear left to right in the same order as they appear top to bottom here.

- **Remove.**
This button removes a field from the list
- **Sort On.**
Check this box if the list is to be sorted on this column. Note that you should only check the box for one column otherwise the result will be undetermined.
- **Reverse.**
Check this box if you want to reverse the sort order.

In the example shown above the Batch Chooser will show all batches that occurred after 20-August-2007, showing the most recently started batches at the top of the list as can be seen below.



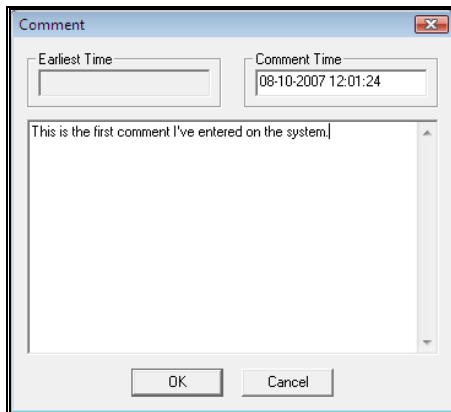
You can change the filtering “on the fly”. Type in any alterations or use the **Advanced** button to display the SQL construction dialog described earlier. After making a change to the filter click the Filter button to make it active. The example below is as above but with the date changed to 11-September-2007.



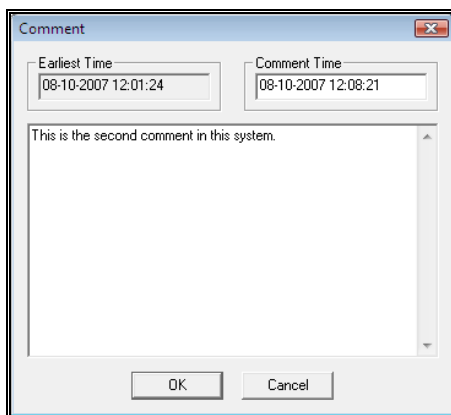
Comments

Comments

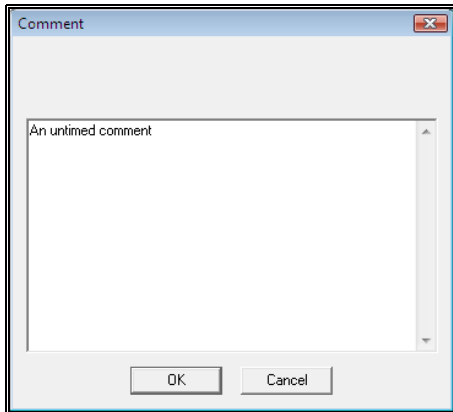
During the process cycle the operators may be required to make certain observations or take manual readings. It could be that this forms a vital part of the QA procedure. Prodigy provides a commenting facility that allows any observations to be entered and timestamped. When viewing historical Trends the positions of any comments may be overlaid on the graph and the comment viewed by clicking on its icon - see the relevant section within Trending (Historical Trending).



The **Comment** program is invoked from the **Menu Bar** or a **Display Builder** button and appears as shown above. The **Comment Time** field is where the operator enters the timestamp that is required for the comment being entered. This field initially shows the current time. The time entered into this field cannot be earlier than the time of the last comment that was entered. As an aid, the time of the last comment entered is shown to the right of the **Comment Time** field and is labelled **Earliest Time** (below).



In some circumstances it may not be desirable for the operator to be able to enter the time of the comment. Suppose for instance that the QA procedures insist that some checks are made at a given time and a comment must be entered at this time to indicate the results of the checks. Disabling the Comment Time field prevents the operators from falsifying the times at which checks are made (below).



To disable the Comment Time field, add the parameter **-notime** in the relevant part of the **Menu Bar Properties**.

Profile Curves

Purpose of Profile Curves

There are two purposes for profile curves within Prodigy. The first is to define and display an “ideal” curve that a signal value should follow, this curve being displayed on a Trend graph alongside the actual trace of the signal for comparison purposes.

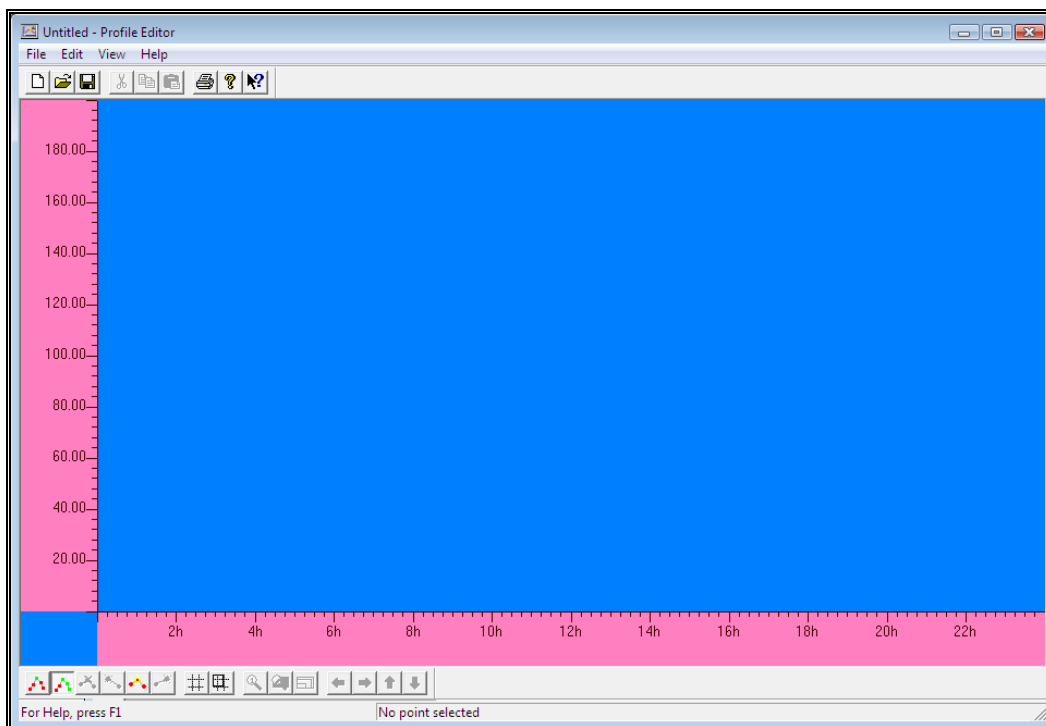
The second use is to prepare control profiles that can be subsequently downloaded into PLCs or standalone three term controllers.

Preparing A Profile Curve

The profile curve is read from a CSV (comma separated variable) file in which the first column specifies the elapsed time in seconds (but see "The CSV File") and the second column specifies the signal value. This file may be generated by hand or by any suitable piece of software – in fact there is a facility within Prodigy Trends to save a graph as a profile curve. Finally, a profile curve may be generated from scratch by using the Profile Editor facility of Prodigy.

The Profile Editor

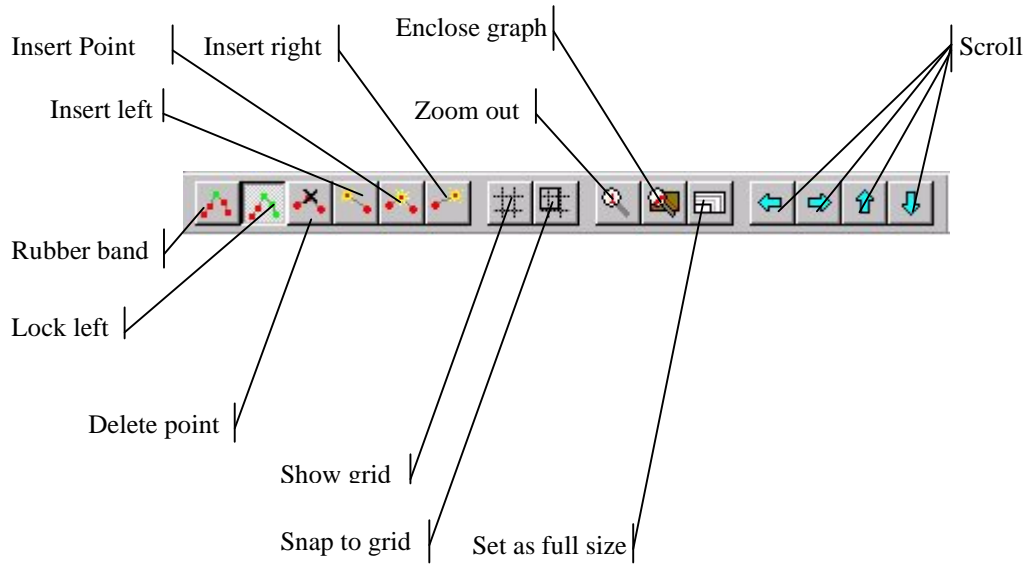
The Prodigy Profile Editor is a graphical editor that provides facilities to create and modify profile curves and to set certain parameters that may be required when producing PLC profiles.



The Profile Editor screen has user-definable axes for elapsed time (X) and value (Y). There is a toolbar at the bottom of the screen that gives access to the functions used to define and modify the profile. Also at the bottom of the screen there is a status bar that displays information about the currently selected point on the profile. At the top of the screen is a toolbar giving access to functions such as save, open and print. Many of these functions are also available as menu items.

The Profile Toolbar

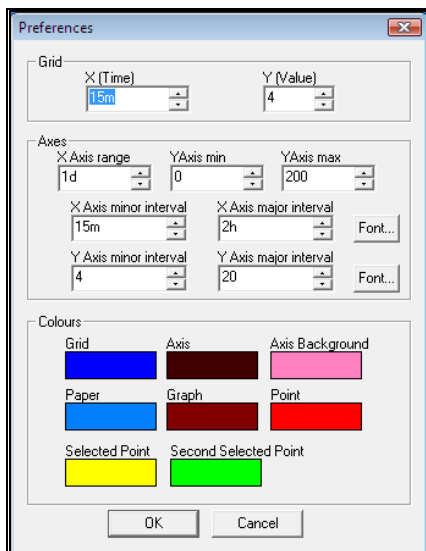
This toolbar gives access to most of the functions needed to create and modify the profile curve.



The meaning of each of these items will be explained in due course. Not all of the options are available at all times. Those that are not available are displayed as being “greyed out” and clicking the option on the toolbar will have no effect.

Screen Properties

There are many aspects of the screen display for which properties may be set. To access the dialog which controls these properties, right click over any part of the central graph area, but not on any part of the curve. The following dialog will be displayed.



- **Grid.**
 - **X (Time)**

Specifies the distance between grid lines in the x direction as a standard Prodigy time period. It is common practice to have this value coincide with the minor tick intervals of the X-axis.
 - **Y (Value)**

Specifies the distance between grid lines in the y direction in arbitrary units. It is common practice to have this value coincide with the minor tick intervals of the Y-axis.
- **Axes.**
 - **X-Axis range.**

Specifies the amount of elapsed time that the graph will display. The origin of the X-axis is always 0 and the right hand extremity will be whatever is entered here. Enter a value in the form of a standard Prodigy time period.
 - **Y-Axis min.**

This specifies the minimum value of the y-axis. This will often be zero, but can be any value including negative values. The value is in arbitrary units.
 - **Y-Axis max.**

This specifies the maximum value of the y-axis. The value is in arbitrary units.
 - **X-Axis minor interval.**

This specifies the distance between minor tick marks on the x-axis specified as a Prodigy time period.
 - **X-Axis major interval.**

This specifies the distance between major tick marks on the x-axis specified as a Prodigy time period.
 - **Font.**

Selects the font to be used when displaying the values of the x-axis.
 - **Y-Axis minor interval.**

This specifies the distance between minor tick marks on the y-axis specified as a Prodigy time period.
 - **Y-Axis major interval.**

This specifies the distance between major tick marks on the y-axis specified as a Prodigy time period.

- **Font.**
Selects the font to be used when displaying the values of the y-axis.
- **Colours.**
The buttons in this group allow the colours of all major aspects of the profile editor to be set. Click on one of the colour buttons to display the colour picker dialog.

Defining A New Profile Curve

When starting with a “blank sheet of paper” the first thing you must do is set the range, both horizontal and vertical that the curve will occupy. Use the Preferences dialog described above to set suitable ranges for the x and y axes and choose suitable tick and grid intervals based on these ranges.

Next you must create a number of points. To create the first point, click the **Insert Point** button. A new point will be created in the centre of the graph area. Click it again and a second point will be created above and to the right of the first point.

Click on the first (leftmost) point and drag it to the position at which the curve is to start. Use the status bar to see the point’s current position. You may want to turn on **Snap to grid** from the toolbar in order to aid positioning the point. Notice that the rightmost point also moves as the leftmost point is moved. To eliminate this action, click the **Rubber band** button on the toolbar. In this mode, whenever a point is moved *only* that point is moved and all others remain stationary.

You can set the position of the point *directly* by right clicking on the point to display the **Point Properties** dialog.

Now select the rightmost point and drag it to the position at which the curve is to end. Of course this just gives you a straight line between two points, which is probably not a lot of good. You must now insert more points in order to build up the curve.

Click on the **Insert left** or **Insert right** button to add a point to the left or right of the currently selected point. Repeat this several times to create several new points. Drag these points to their required position or set their positions using the **Point Properties** dialog.

Modifying a Curve

A newly created curve or one that has been read in from a file can be modified by dragging the points to new positions or by setting their positions by using the **Point Properties** dialog. When moving a point the movement can be constrained to be either horizontal or vertical.

- To move a point vertically, press the **shift** key while moving the mouse.
- To move a point horizontally, press the **control** key while moving the mouse.



When moving a point in **Rubber band** mode the point cannot be moved to a time earlier than the previous point or later than the next point.



When moving a point in **Lock left** mode all of the points to the right of the current point move with it. All of the points to the left remain stationary and the current point cannot be moved to a time that is earlier than that of its preceding point.



To remove superfluous points, select the point and press the **Delete point** button or the **del** key.



To insert new points press the **Insert left** key to insert a point midway between the current point and the point to its left.



To insert new points press the **Insert right** key to insert a point midway between the current point and the point to its right.



To insert a new point between two points select both points and click the **Insert Point** button.

Zooming

It is often desirable to zoom in on a portion of a curve for greater clarity. To do this, use the left mouse button to drag a box over the area you want to zoom in on. A dialog will appear asking you to confirm that you want to zoom in. If a positive reply is given the curve will be redrawn to the new larger scale.



To zoom back out, press the **Zoom out** button on the toolbar to restore the original scale. Note that no matter how many times you zoom in, **Zoom out** always comes back to the original scale defined in the **Preferences** dialog.



Another way of zooming is to press the **Enclose graph** button. This recalculates the axes to enclose all of the points making up the current profile. A 10% margin is added to each side of the Y-axis and to the rightmost extremity of the X-axis. **Zoom out** will still return to the original scale.

Rescaling



After zooming you can set the new scale to be the actual size of the portion of graph shown by pressing the **Set as full size** button on the toolbar. This effectively means that you are no longer zoomed in but are now at the full working scale for this profile. The **Zoom out** button will now be greyed out.

Scrolling

Scrolling can only be performed when zoomed in on the curve. The arrow buttons on the toolbar allow the image to be scrolled left, right, up or down by an amount equal to half of the viewed size.

Point Properties & Annotation

Right clicking over a point displays the **Point Properties** dialog.

Point Position	
Time	Value
48s	240.53191489

Annotation	
Label	Length to left
	0s
<input checked="" type="radio"/> Right	Length to right
<input type="radio"/> Above	0s

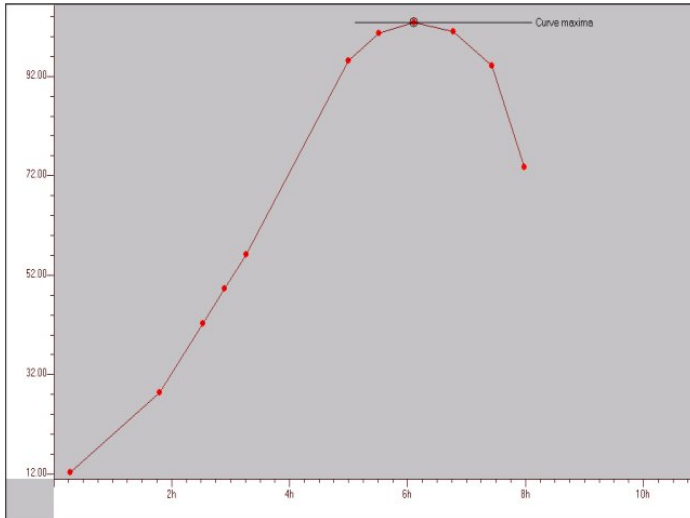
Controller Parameters

OK Cancel

This dialog performs three tasks.

- Firstly, it allows the exact position of the point to be specified. This provides for more accurate positioning than dragging may allow, when circumstances demand this.
- Secondly, it allows a point to be annotated. A line is drawn either horizontally or vertically through the point, depending on whether **Right** or **Above** is selected and the Label is drawn to the right or at the top of this line. The amount by which the line extends to the left and right of the line (or above

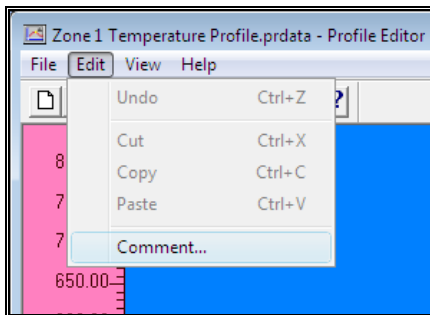
and below the line) is specified in the two boxes at the right of this section of the dialog. When specifying the distance to the left and right the value must be entered as a Prodigy time period.



- Finally, it allows values to be set for certain parameters that may be passed to a PLC if this curve is downloaded. Whether or not this final function is available and which parameters can be set depends on the **Device** currently selected for this curve. The device is selected by using the **Select Device...** option from the **File** menu.
Currently there are no devices implemented.

Comment

A comment can be added to a profile curve by selecting **Comment...** from the **Edit** menu. This displays a dialog containing the current comment (if any) and allows a new comment to be entered. This comment is stored in the CSV file along with the data.



The CSV File

The format of the CSV file is as follows.

Time, Value, Label, Above, Distance1, Distance2, Analogue1, Analogue2, Analogue3, Analogue4, Digital1, Digital2, Digital3, Digital4

The values for Label, Above, Distance1 and Distance2 are used to hold any annotation that may occur at a point. Analogue1 to 4 and Digital1 to 4 are the various parameters used in PLC download. Only the Time and Value pairs are mandatory when creating a CSV file by hand or from other software.

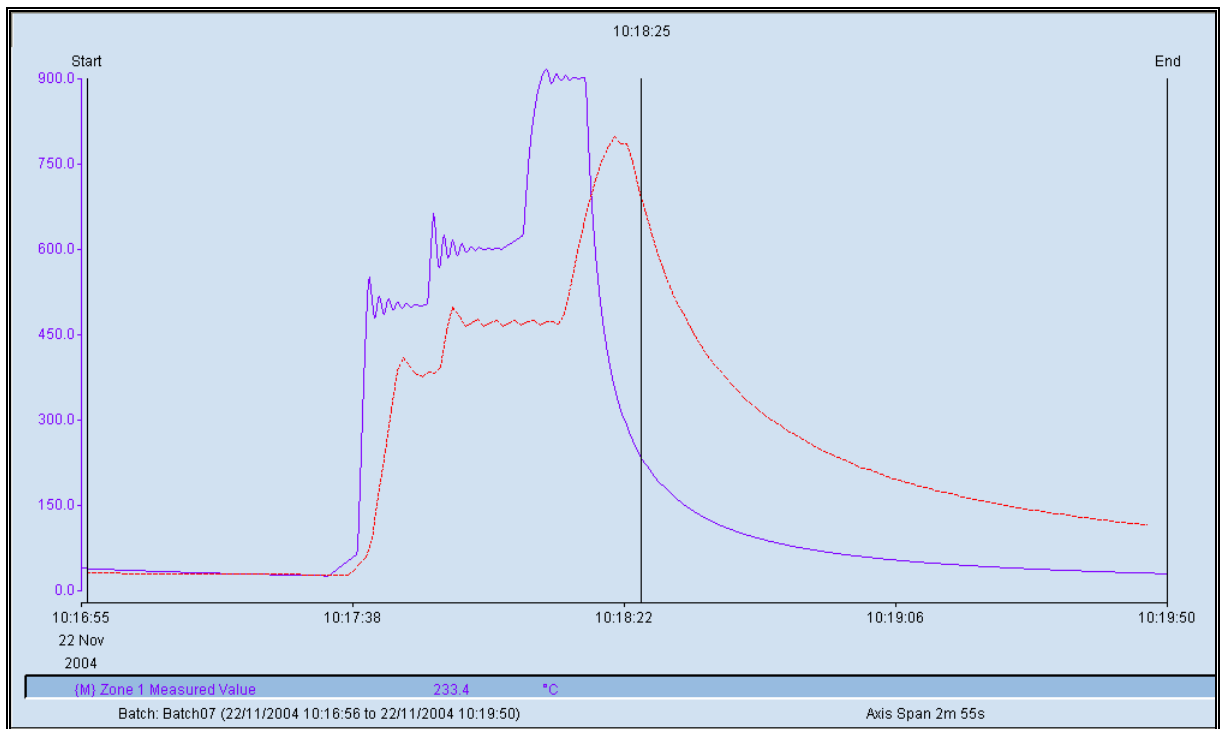
Additionally, the following special symbols and keywords may appear in the CSV file.

- **! Config=<filename>**
specifies the path of the device description file for the PLC being used. Not implemented yet.

- **! Timescale=S**
S may be replaced by M, H or D and indicates that the times given in the first column are in seconds, minutes, hours or days respectively.
- **! Comment=Text**
Text is replaced by the actual comment. Many consecutive lines of comment may be specified, each starting with ! Comment=
- *****
Any line beginning with * is ignored when the file is parsed and will be lost if the file is read and saved by Profile Editor.

Using the Finished Profile Curve

Profile curves can be added into a Trend in Prodigy Display Builder. They are displayed when the Trend is viewed in historical mode and can be used to compare against a trace on the trend.



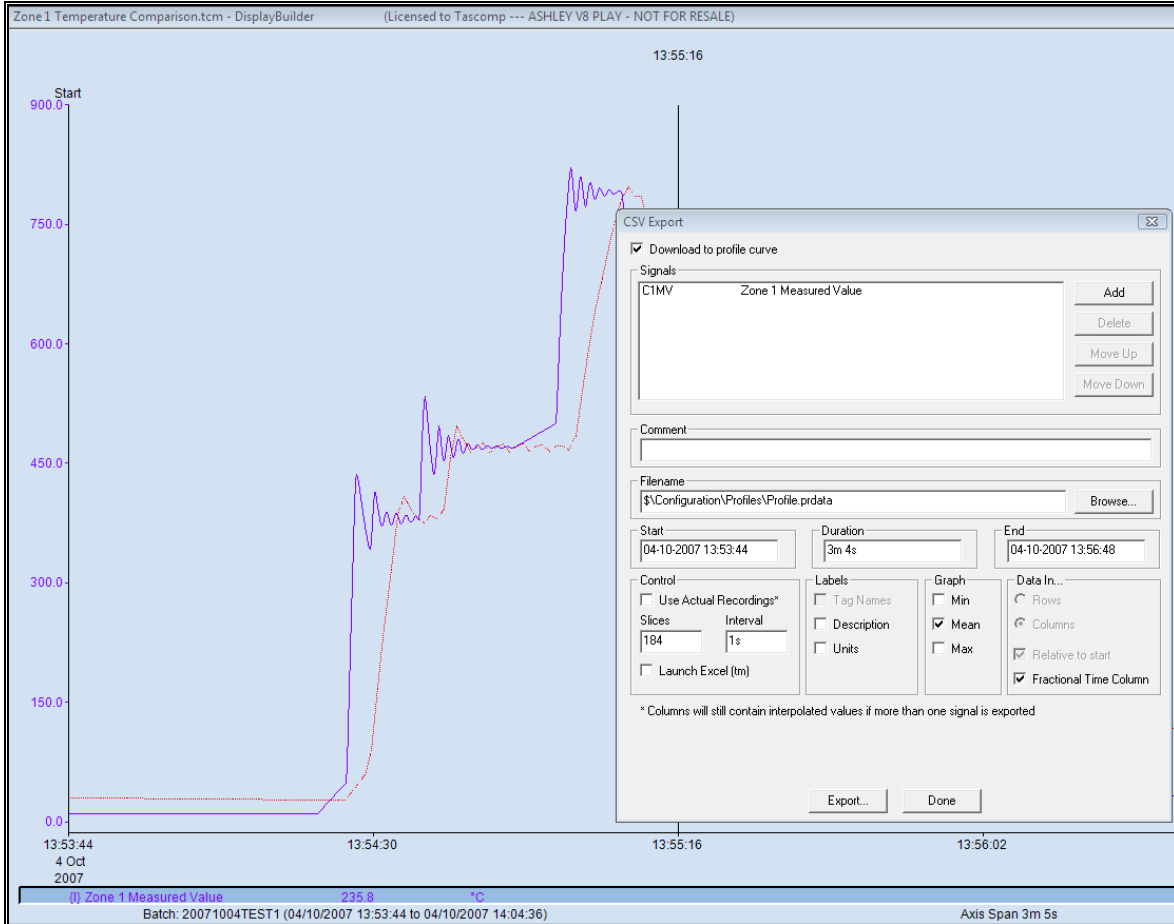
The red trace is the ideal curve used throughout this chapter, the purple trace is one from a batch run of Alchemy.

Exporting Curves From A Trend

You can create an Ideal Curve by exporting a curve from a trend and using it as-exported, or tweaking it first in the Profile Editor.



Simply select the CSV Export button in Historical Trending and check the Download to profile curve box at the top of the dialog. Enter a sensible value for the number of slices – too many will be unworkable, too few will lose accuracy.



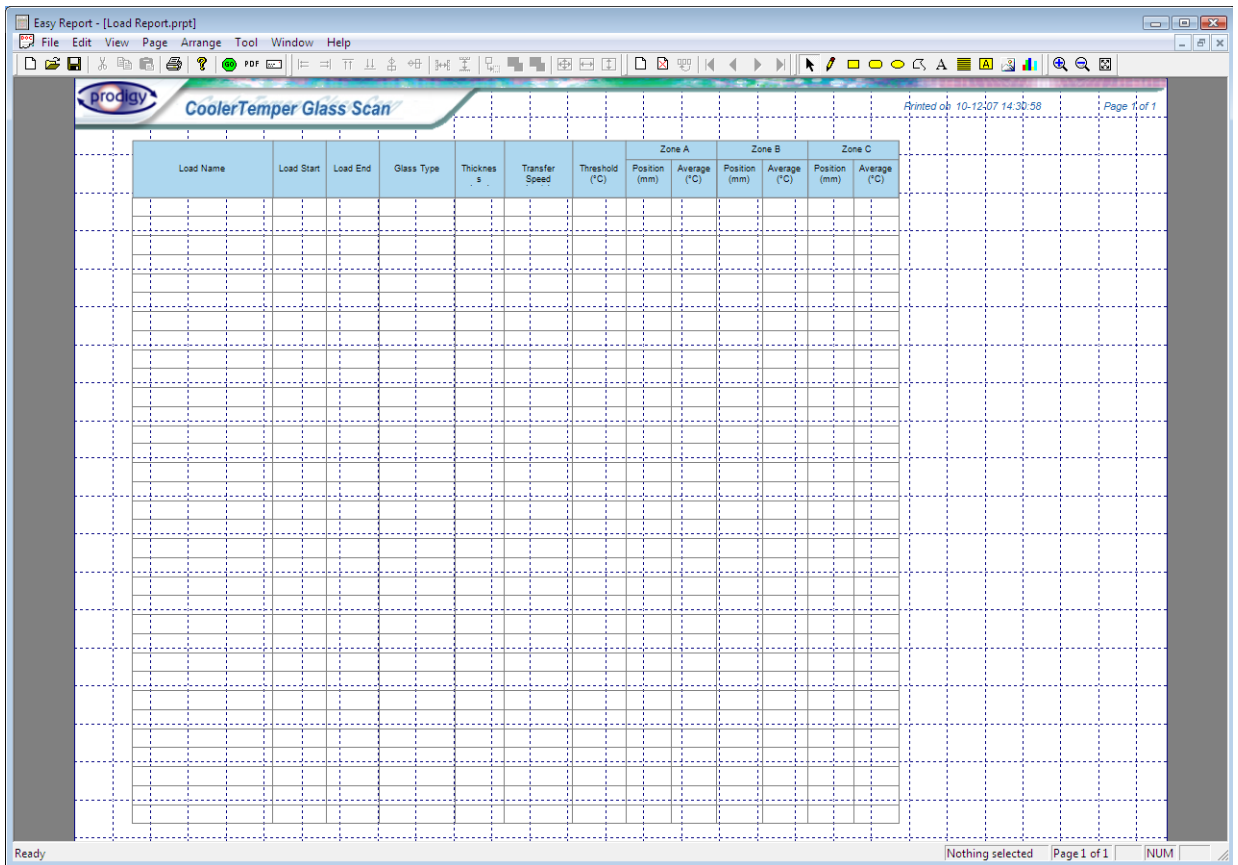
Easy Report

Easy Report is a tool that can be used to produce either tabular or graphical reports from Batch or Engineering Test data. A typical report will look like the illustration below.

Batch Name	Batch Start	Batch End	Duration	Glass Type	Thickness (mm)	Number of Loads	Loads (per hour)	Zone A		Zone B		Zone C	
								Position (mm)	Average (°C)	Position (mm)	Average (°C)	Position (mm)	Average (°C)
12345678_Type 1_07_Jan_05_10_	10:18:19			Type 1	6			1200		2400			3600
500T_Type 2_10_Dec_04_16_06	16:06:23			Type 2	6			1200		2400			3600
ASQW123_Type 2_10_Dec_04_15	15:06:45	16:06:13	0h 59m 28s	Type 2	6	25	25.2	1200	40.2	2400	44.6	3600	36.9
FRED_Type 2_10_Dec_04_14_08	14:08:27	14:11:36	0h 3m 9s	Type 2	6	4	76.2	1200	40.3	2400	37.9	3600	36.3
Total			1h 2m 37s			29							
Mean							50.7		40.2		41.3		36.6

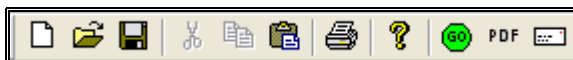
Using Easy Report












When you run Easy Report you will notice three main areas (below): the menu bar, the tool bar and the main work area where you lay out your report.



The Tool Bars




Main Toolbar



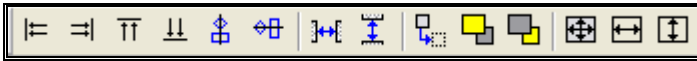
- **New.**
 This creates a new Easy Report document. You may have multiple Easy Report documents open at once within Easy Report.
- **Open.**
 This opens an existing Easy Report document.
- **Save.**
 This saves the active Easy Report document.
- **Cut.**
 This cuts the currently selected object(s) from the document and places them in the clipboard.
- **Copy.**
 This copies the currently selected object(s) from the document and places them in the clipboard.
- **Paste.**
 This pastes any objects from the clipboard into the active document.
- **Print.**
 This prints the active report to the selected printer.
- **Help.**
 This displays information about the version of the reporter.
- **Go.**
 This runs the active report in Print Preview mode. A dialog will be displayed that requests either the batch(es) to report on or the dates between which the report is to be generated, depending on how it has been configured. When the report runs, feedback dialogs may display the current action (e.g. Producing Temporary Database).
- **PDF.**
 This produces a PDF format file of the active report (requires PDF facility in your license).
- **Email.**
 This produces a PDF format file of the active report and emails it to a Prodigy user (requires both the PDF facility and the email facility in your license).

View Toolbar



- **Zoom In.**
 Zooms into the selected area. Select an area by clicking and dragging on the display while the Zoom In tool is active.
- **Zoom Out.**
 Zooms out when clicked on the visible area.
- **Zoom Full.**
 Sizes the zoom to fit the entire document on the screen.

Arrange Toolbar



Some of these tools work on multiple selected objects. To select multiple objects, either drag a marquee around the objects you wish to select using the Select tool, or add/remove objects to/from the selection by holding down the shift key and clicking on the required object. The last selected object will have black handles around it, with the other objects having blue handles.

- **Align Left.**



Arranges the left edges of all the selected objects to align with the left edge of the last selected object.

- **Align Right.**



Arranges the right edges of all the selected objects to align with the right edge of the last selected object.

- **Align Top.**



Arranges the top edges of all the selected objects to align with the top edge of the last selected object.

- **Align Bottom.**



Arranges the bottom edges of all the selected objects to align with the bottom edge of the last selected object.

- **Align Vertical Centre.**



Arranges the selected objects along the vertical centre of the last selected object.

- **Align Horizontal Centre.**



Arranges the selected objects along the horizontal centre of the last selected object.

- **Duplicate.**



Duplicates the currently selected objects.

- **Bring To Front.**



Brings the selected object to the top of the document.

- **Send To Back.**



Sends the selected object to the bottom of the document.

- **Make Same Size.**



Re-sizes the selected objects to be the same size as the last selected object.

- **Make Same Width.**



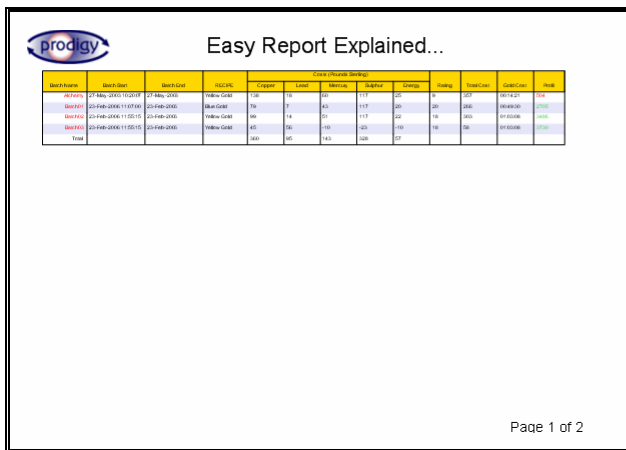
Re-sizes the selected objects to be the same width as the last selected object.

- **Make Same Height.**

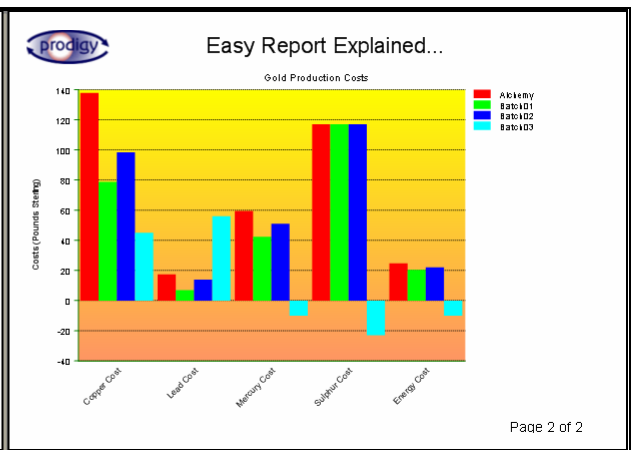


Re-sizes the selected objects to be the same height as the last selected object.

The output will look like this:




Page 1 of 2





Page 2 of 2


In practice when the report is run the data block may itself take up more than one page, depending on how much data there is, but the barchart will still appear on the next page after the data block ends, and so on.


Items like the bitmap, header and footer in this example as well as drawing items (lines, rectangles, etc.) can be configured to appear on all pages or just the current page. See the individual items (explained later) for more details.


 To create a new page click the **New Page** button. The new page goes at the end of the document but you can reposition it later.


 To delete the current page click the **Delete Page** button.

 Move a page to a different position by clicking the **Move Page** button. This displays a dialog that asks for the new page number that the current page is to take. This must be greater than 0 and less than or equal to the total number of pages.

 Display the first page by clicking the **First Page** button.

 Display the previous page by clicking the **Previous Page** button.


 Display the next page by clicking the **Next Page** button.


 Display the last page by clicking the **Last Page** button.


Tool Toolbar



- Select.**

 Use to select objects by clicking, shift-clicking or dragging a marquee around them.
- Line.**

 Click and then drag to draw a straight line. The end handles may be dragged to change the size and orientation. Dragging part of the line that isn't a handle may move the whole line.
- Rectangle.**

 Click and then drag to draw a rectangle. The corner and side handles may be dragged to change the size. Dragging part of it that isn't a handle may move the whole rectangle.

- **Rounded Rectangle.**



Click and then drag to draw a rectangle with rounded corners. The corner and side handles may be dragged to change the size. Dragging part of it that isn't a handle may move the whole rectangle.

- **Ellipse.**



Click and then drag to draw an ellipse. The corner and side handles may be dragged to change the size. Dragging part of it that isn't a handle may move the whole ellipse.

- **Polygon.**



Click to start a polygon, then click again to place each point. Double-click to place the last point (it will connect up with the first point). Each point will have a handle that can be dragged to move it. Dragging part of it that isn't a handle may move the whole polygon.

- **Text.**



Click and then drag to define the area you want the text to occupy. Alter the properties by right-clicking the text and using the dialogs explained below.

- **Data Block.**



A data block is a repeating table of data, similar to a spreadsheet. Click and then drag to define the height of the area you want the data block to occupy (the width is automatically determined by the amount of data you choose to display in the data block). See later for how to configure a Data Block object.

- **Field.**



A field is a single piece of data. Click and then drag to define the area you want the field to occupy. See later for how to configure a Field object.

- **Bitmap**



Click and then drag to define the area you want the image to occupy.

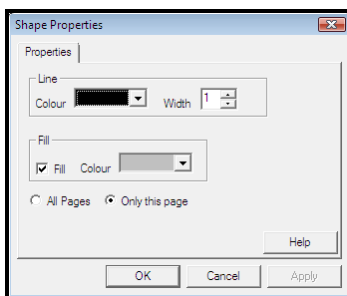
- **Chart**



A chart can be either one of a number of types of bar chart or pie chart. Click and drag to define the area that you want the chart to occupy. See later for how to configure the chart object.

Object Properties

The properties of the above objects can be accessed by right-clicking over them. This will bring up a dialog like the one shown below, for drawing objects (polygon, square etc.):



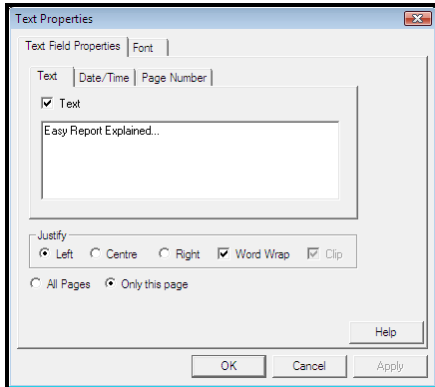
You may change the colour and line thickness of the shape, and, for relevant objects, the fill colour. Choose where you want the object to appear by clicking the **All Pages** or **Only this page** button.

Text Properties

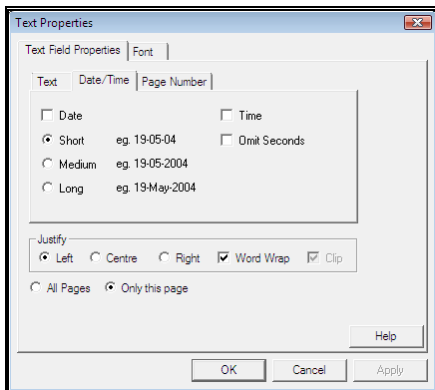
A text item can be one of three types: free format **Text**, a **Date/Time** or a **Page Number**, determined by their own particular tab on the **Text Properties** dialog. For all of these types, you may choose the justification and

whether the words are wrapped within the rectangular area that has been dragged out. Choose where you want the object to appear by clicking the **All Pages** or **Only this page** button.

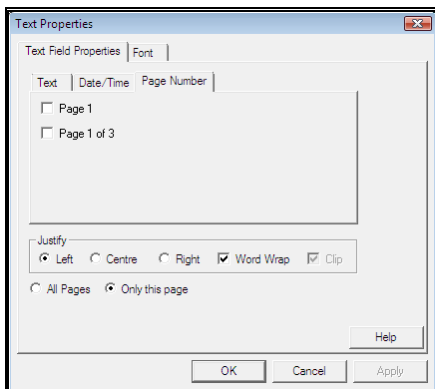
For free format text, enter the text into the Text tab and tick the Text box (below).



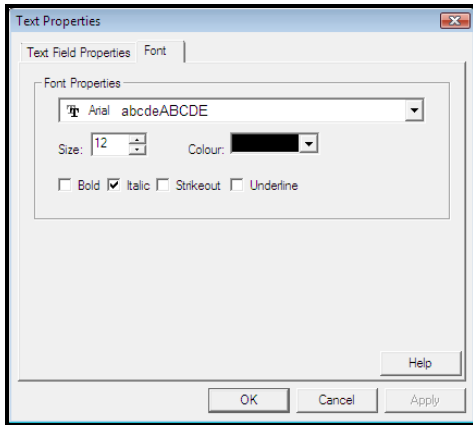
For Date/Time, choose the combination of date and/or time you require, and one of the display formats (Short, Medium or Long). You may choose to omit seconds from the time part (below).



For Page Number, choose to display just the page number or the page number and total number of pages (e.g. Page 1 of 3) (below).



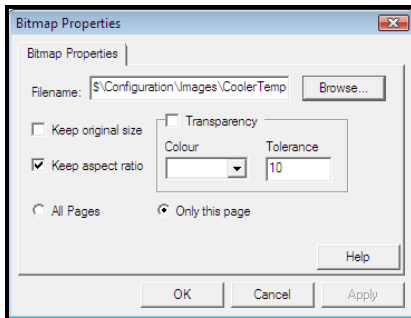
Choose the font, size, style and colour on the Font tab (below).



If you would like to set a default for these so that you don't need to change this every time you create a text object, go to **Edit->Default Font Properties** and set them there.

Bitmap Properties

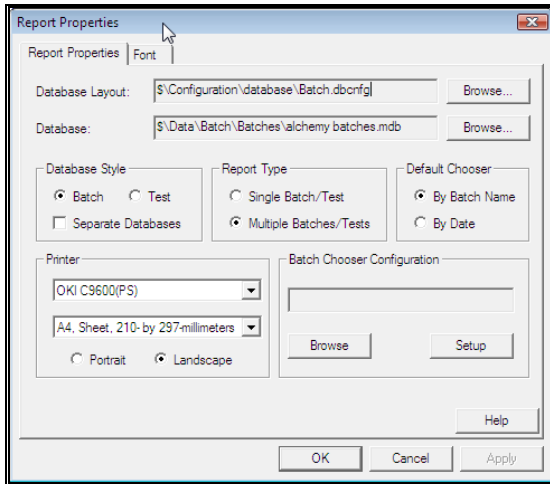
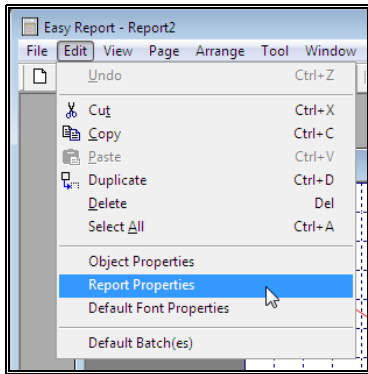
Alter the properties by right-clicking the text and using the following dialog...



- **Filename.**
Pick the image you want to display using the Browse button.
- **Keep original size.**
Constrain the image size so that it can't get altered.
- **Keep aspect ratio.**
Constrain the image dimensions so that the original aspect ratio is kept when re-sizing, preventing distortion.
- **First page only.**
You may opt to display the image on all pages (the default), or just on the first page. Easy Report will automatically resize any Data Blocks to take into account any extra space available on pages after the first one.
- **Transparency.**
You may pick a colour in the image to become transparent. The Tolerance value allows colours similar to the one picked to be transparent also.

Report Properties

Before you can do anything you **must** select a database from the **Edit->Report Properties** menu. While you are doing that you may as well configure all of the Report Properties, as detailed below.



Select the actual database to be used along with the database layout file that defines it (see **Access Database Configuration**, elsewhere in this manual). Decide whether the database is of type **Batch** or **Test** (it will most likely be Batch, Test is a rather specialist area). Also, select whether or not you are using **Separate Databases** (where each batch resides in its own database, rather than all batches in one database). If you are, then the Database path should be just a path, and not an actual file.

Decide if the **Report Type** is to be based on single batches or multiple batches. If multiple is selected then when the report is run you can select multiple batches from the Batch Chooser dialog that is displayed.

You must also decide (in the **Default Chooser** group) if you want to select batches **By Batch Name** or **By Date** range. If you choose the latter, then all batches that fall within the specified (at runtime) date range will be automatically selected.

If you are choosing **By Batch Name** you can also specify a **Batch Chooser Configuration** to use. This allows you to specify which fields are shown in the batch chooser and also enables you to pre-filter the batches that are displayed. See elsewhere in the manual for how to configure a Batch Chooser.

Select the **Printer** that this report is to go to, and also whether it is to be in **Portrait** or **Landscape** orientation.

Configuring A Data Block Object

Data Table Properties Tab

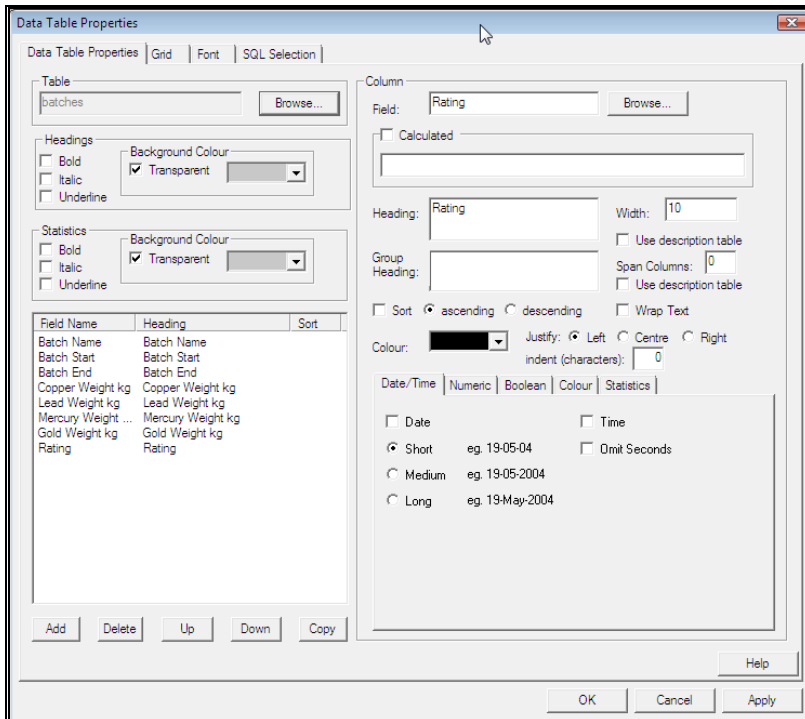


Table.

A data block gets all of its information out of one table in the database. Choose the table to use here.

Headings.

Attributes of the heading text and its background can be chosen here. Make the heading stand out by making it bold etc.

Statistics.

A data block can display summary statistics of the data columns. Change various attributes as for the headings.

Column list.

Lists the fields from your table you wish to display. **Add**, **Delete** and change their display order in the list (**Up**, **Down**). You can also **Copy** a field, which makes an exact copy of the currently selected field and places it directly below (**Adding** fields always places them at the bottom).

Column.

The details of the column that you have currently selected in the column list. The fields within this group are detailed below.

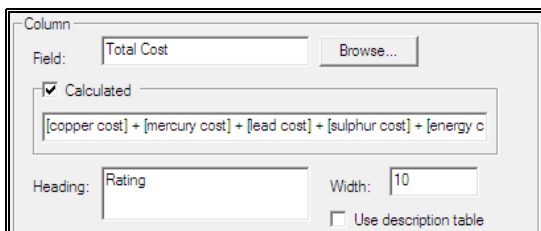
Field.

Choose (i.e. alter) the field that you want to display, either by browsing the available fields in the table, or by typing the field name in manually. This will change the name of the field shown in the Column List too. You will need to enter a name manually for calculated fields, as they will (in fact **must**) not exist in the database. If you give a calculated field a name that already exists in the database table it will result in an error at runtime.

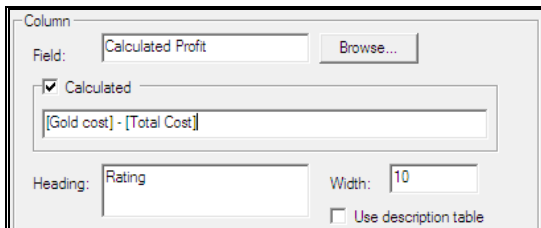
Calculated.

Easy Report can display data that is calculated from other fields, mathematical expressions etc. Any valid SQL calculation can be used, and you can include other fields in the table in your calculations.

To use a calculated field first create a field by adding any field (it doesn't matter what it is) to the **Column List**. Click on that field in the Column List and then check the **Calculated** box. Give your calculated field a name by typing it into the **Field** box (the name **must** not already exist in the database table) and then enter the required expression into the **Calculated** field. When using field names within the calculation, enclose them in square brackets. See the example below.



You can use calculated fields within other calculations. The only limitation is that a calculation used within another must have already been calculated (i.e. it must have been higher in the Column List).



The calculation that you enter must be valid, i.e. it must contain parts that can logically go together in a calculation. For instance you cannot divide a number by a string and so on. If your equation is invalid it will not yield a result when it is executed and an error may appear in **Prodigy Spy**.

When using Date/Time fields (i.e. fields which represent an actual date/time rather than a duration) within a calculation the Date/Time field is treated as a decimal number of days since 30 December 1899. To use this on its own is pretty meaningless, but if you do a subtraction of two date/dime fields you will get the difference between them in decimal days (e.g. 3.25 = 3 days and 6 hours).

Tip: Use Microsoft Access and try out your SQL expressions to make sure they will work on your database.

Heading.

The text label to display above the column. By default it is the same as the field name when a field is selected from a table.

Width.

The number of characters to reserve for the column.

Use description table.

Tick this check box to get the heading from another table within the database. It is only used for specialised down time monitoring applications and is not generally applicable.

Group Heading.

A group heading is positioned above a normal heading, and may span multiple columns e.g. a group heading of “Weight (kg)” could cover the columns for “Copper”, “Lead”, “Mercury” and “Gold”. You enter the name of the group heading in the first column that is part of the group and specify how many columns (including this first one) that it is to span.

The dialog box contains the following fields and options:

- Heading: Width:
- Group Heading: Span Columns:
- Use description table (next to both Heading and Group Heading)
- Sort ascending descending Wrap Text
- Colour:
- Justify: Left Centre Right
- indent (characters):

This produces output as shown below.

RECIPE	Weight (kg)				Rating
	Copper	Lead	Mercury	Gold	

Sort.

The rows may be sorted on either ascending or descending order of the data within them. Sorting may be combined across columns, with the left-most columns having a higher priority.

Colour.

The colour of the text displayed in this column (but not including the header).

Justify.

Choose the alignment of the data. For numeric values, right-justification works best.

Indent.

The value entered here causes the value in the column to be indented by that number of characters. This applies to Left aligned columns only and should not be used with other types of justification.

Depending on the type of data that will be displayed in the column, you may configure its appearance in the following tabs:

Date/Time Tab.

The dialog box has tabs for Date/Time, Numeric, Boolean, Colour, and Statistics. The Date/Time tab is active and contains:

- Date Time
- Short eg. 19-05-04 Omit Seconds
- Medium eg. 19-05-2004
- Long eg. 19-May-2004

For a field that is a date/time, you may choose to display the date, the time or both. The date may be displayed in one of three formats, examples of which are given on the dialog.

Numeric Tab.

For numeric fields, choose the number of decimal places and whether to omit zero values (i.e. leave the cell blank). You may wish to omit zero values to make them stand out more clearly, or because a large report with many zero values in it can make it harder to see the other values which are of more interest.

A numeric field may be treated as a time if it represents a count of seconds: choose whether to interpret it as a count of days, hours, minutes or seconds. You can choose any combination so long as you don't leave gaps (e.g. don't pick Hours and Seconds without picking Minutes too). The time is displayed as a time period (e.g. 4h 31m 20s) unless you check the **Short form** box, which will then display it in normal time format (4:31:20) (see below).

The screenshot shows the 'Numeric' tab configuration window. At the top, there are tabs for 'Date/Time', 'Numeric', 'Boolean', 'Colour', and 'Statistics'. Under 'Decimal Places', there is a text input field with '0' and a checkbox for 'Omit zero values'. The 'Time' section has checkboxes for 'Treat as time', 'Short form', 'Day', 'Hour', 'Min', and 'Sec'. The 'List' section has checkboxes for 'Treat as list' and 'Use signal', and a 'Browse...' button. Below this is a table with columns 'Value' and 'Text', and buttons 'Add', 'Delete', and 'Edit'.

Numeric fields may also be interpreted as a list. For instance, if the field uses different numeric values to represent textual values, such as 1=Red, 2=Blue, 3=Green, you can specify that here. If the signal is a Finite State then you can use the signal's own event texts by clicking the **Use Signal** box. Alternatively you can specify your own text by entering values into the Value/Text mapping list by using the **Add**, **Del** and **Edit** buttons alongside the list (see below).

The screenshot shows the 'List' configuration window. It has checkboxes for 'Treat as list' (checked) and 'Use signal' (unchecked), and a 'Browse...' button. Below is a table with columns 'Value' and 'Text', and buttons 'Add', 'Delete', and 'Edit'. The table contains the following data:

Value	Text
1	Red
2	Blue
3	Green

Boolean Tab.

The screenshot shows the 'Boolean' tab configuration window. It has input fields for '"True" text:' and '"False" text:'. There is also a checkbox for 'Use signal' and a 'Browse...' button.

Boolean values (those that can either be 0 or 1) may be displayed as text. Either enter the text that you want to be used for 0 and 1 or else use the event text values from the signal within Prodigy by clicking the **Use signal** box.

Colour Tab.

The colour of the value in question may be varied with the value of any field on its row (including itself). Choose up to 5 different colours, and the range of values associated with them. This may, for example, be used to display a value in Green if it is deemed as being “good” and in Red if it is “bad”, where good and bad are determined by the ranges specified (see below).

Curry	Gold	Rating	Total Cos	Gold Cos	Profit
100	9	357	00:14:21	504	
345	20	266	00:49:30	2705	
440	18	303	01:03:08	3486	

Statistics Tab.

Any numeric field may have summary statistics of Total, Max, Min or Mean displayed at the end of the report. Choose which of these you want to include (you can have any combination of them), and the number of decimal places required. See below.

Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Alchemy	27-May-2003 10:20:	27-May-2003	Yellow Gold	70	50	70	100	9	357	00:14:21	504
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486
Total											6695
Mean											2232

Grid Tab

A data block can be displayed within a grid to aid its readability. There are various options to customise the grid’s appearance, such as displaying horizontal lines of a given style and colour and shading alternate rows. Some of these features can be seen in previous example output, but below is a small selection of what can be achieved.

Just grid lines:

Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Alchemy	27-May-2003 10:20:	27-May-2003	Yellow Gold	70	50	70	100	9	357	00:14:21	504
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486

No grid lines:

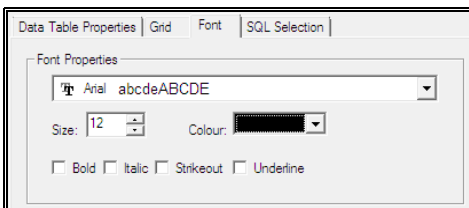
Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Alchemy	27-May-2003 10:20:	27-May-2003	Yellow Gold	70	50	70	100	9	357	00:14:21	504
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486

Alternate shading:

Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Alchemy	27-May-2003 10:20:	27-May-2003	Yellow Gold	70	50	70	100	9	357	00:14:21	504
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486

N.B. The display of Line Style on a printer is dependent on the manufacturer's printer drivers, and some printers do not support the different line styles.

Font Tab.



Choose the font attributes to be used for the data. The size of the font and the number of characters that you choose to display across the columns determine the width of the Data Block.

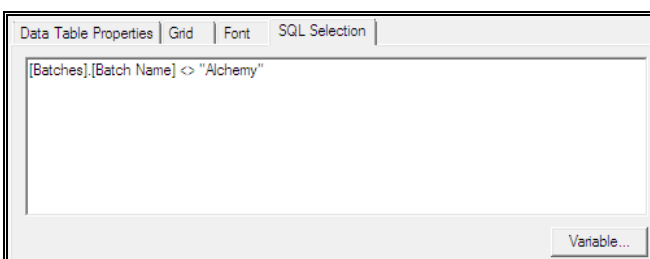
SQL Selection Tab.

You can specify an SQL string that will perform further filtering on the data that you select. No matter whether you select your data by dates or by batches, the SQL Selection string will be applied to all of the data collected when the report is run and perform further filtering. The SQL string entered will be used in the WHERE part of the SQL statement "Select * from <database> WHERE..." An example is given below.

With no SQL Selection:

Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Alchemy	27-May-2003 10:20:	27-May-2003	Yellow Gold	70	50	70	100	9	357	00:14:21	504
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486

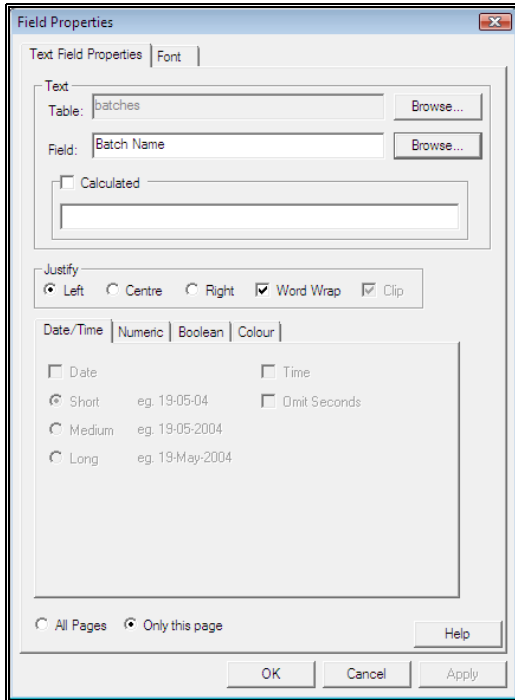
SQL Selection:



After SQL selection:

Batch Name	Batch Start	Batch End	RECIPE	Weight (kg)				Rating	Total Cos	Gold Cos	Profit
				Copper	Lead	Mercury	Gold				
Batch02	23-Feb-2006 11:55:	23-Feb-2006	Yellow Gold	50	40	60	440	18	303	01:03:08	3486
Batch01	23-Feb-2006 11:07:	23-Feb-2006	Blue Gold	40	20	50	345	20	266	00:49:30	2705

Configuring A Field Object



Choose the data you wish to display by browsing the **Tables** and **Fields** within your database. You may opt to display the field on all pages (the default), or just on this page by checking the **All Pages** or **Only This Page** button.

Date/Time Tab.



This is exactly as in the **Data Block** described earlier.

Numeric Tab

The screenshot shows a configuration window with tabs for Date/Time, Numeric, Boolean, and Colour. The Numeric tab is active. It includes a 'Decimal Places' field set to 0 and an 'Omit zero values' checkbox. A 'Time' section has checkboxes for 'Treat as time', 'Short form', 'Day', 'Hour', 'Min', and 'Sec'. A 'List' section has checkboxes for 'Treat as list' and 'Use signal', a 'Browse...' button, and a table with 'Add', 'Delete', and 'Edit' buttons. The table has columns for 'Value' and 'Text'.

This is exactly as in the **Data Block** described earlier.

Boolean Tab

The screenshot shows a configuration window with tabs for Date/Time, Numeric, Boolean, and Colour. The Boolean tab is active. It includes fields for '"True" text:' (set to True) and '"False" text:' (set to False). There is a 'Use signal' checkbox and a 'Browse...' button.

This is exactly as in the **Data Block** described earlier.

Colour Tab

The screenshot shows a configuration window with tabs for Date/Time, Numeric, Boolean, and Colour. The Colour tab is active. It includes a 'Vary colour with value of:' field and a 'Browse' button. Below are five rows, each with a 'Use' checkbox, a color selection dropdown (all set to black), the word 'between', and two numeric input fields (both set to 0) separated by 'and'.

This is exactly as in the **Data Block** described earlier except that the colour varies by any other **Field Object** (including this one) rather than an item in a row.

Configuring A Chart Object

Bar Chart Properties

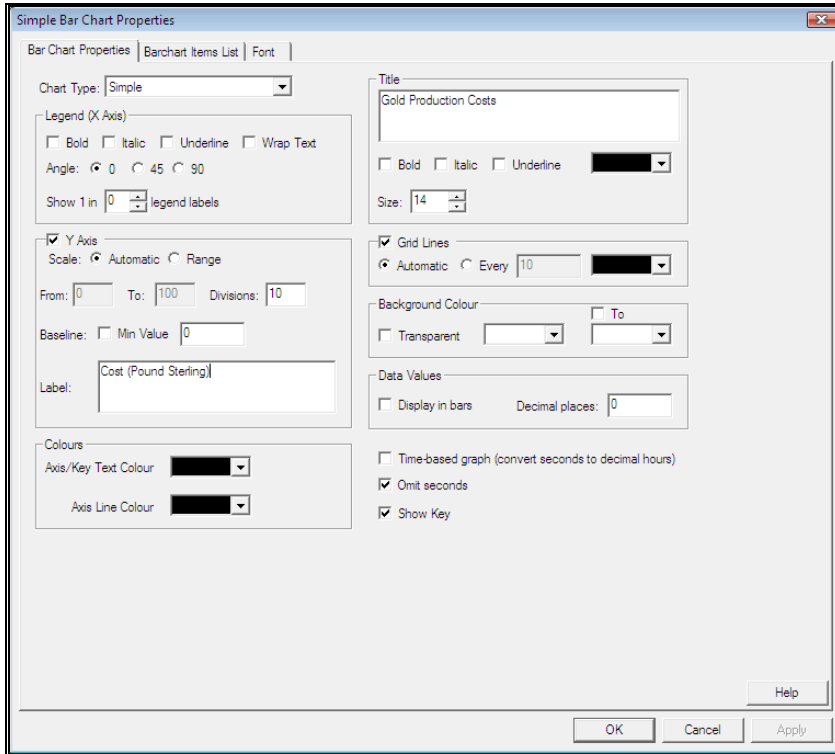
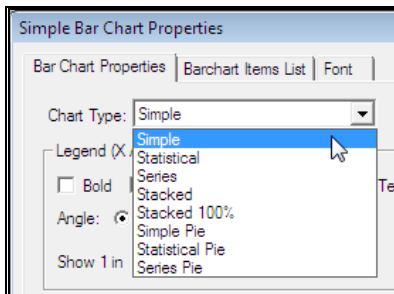


Chart Type



There are several chart types available – namely a variety of barchart and pie charts. Each chart type is covered in more detail later, along with examples of how the output looks for each type of chart.

Legend (X Axis)

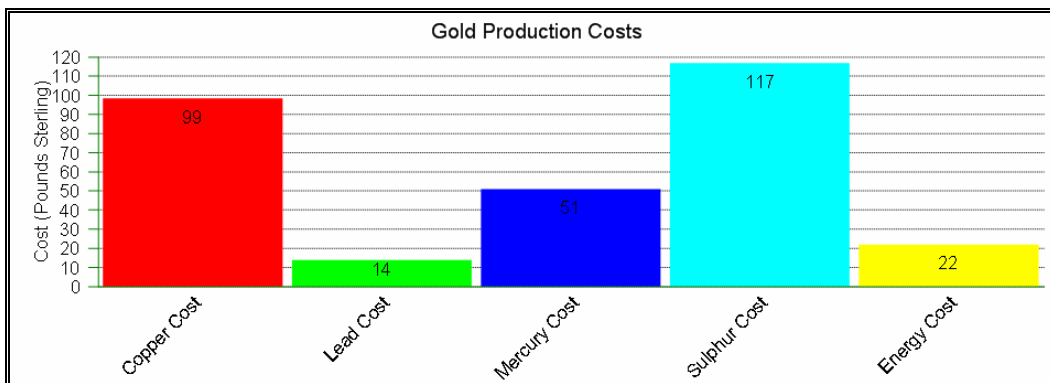
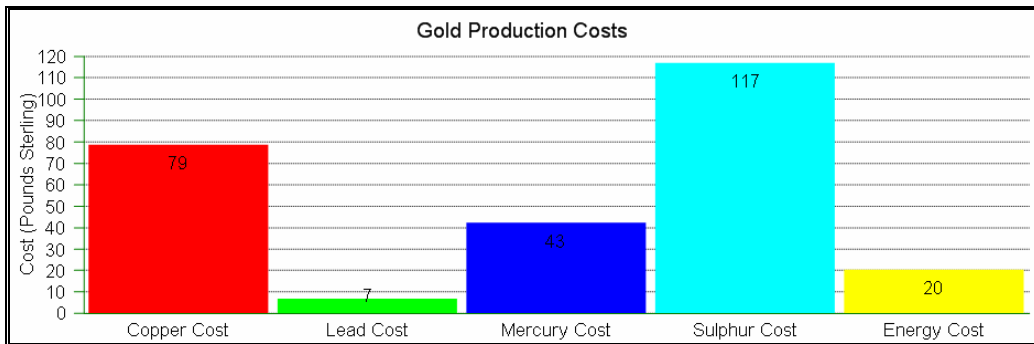
Legend (X Axis)

Bold Italic Underline Wrap Text

Angle: 0 45 90

Show 1 in legend labels

Select the text colour and style for the X Axis labels (or Legend), along with the angle at which they are to be drawn. 90 degrees is best used when there are a lot of bars to be displayed, otherwise the labels will clash into each other. Even at 90 degrees the labels could still clash, so the **Show 1 in** box can be used to limit the number of labels that are displayed (selecting 1 in 2 will cause every second label to be displayed, and so on).



Y Axis

Y Axis

Scale: Automatic Range

From: To: Divisions:

Baseline: Min Value

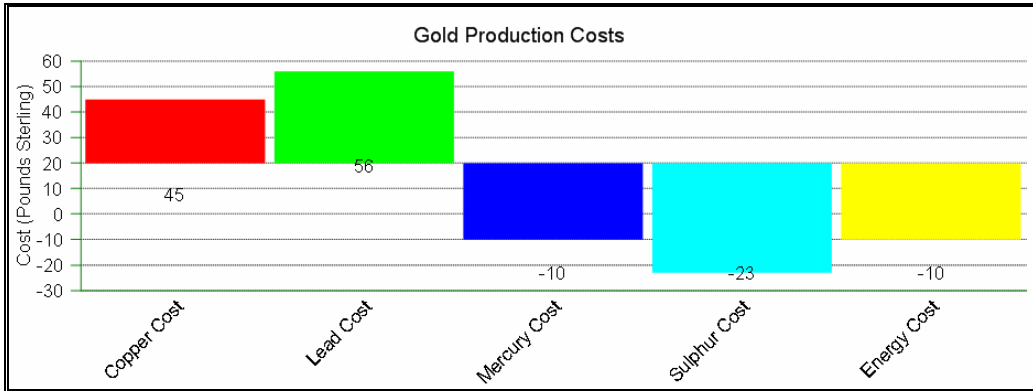
Label:

You do not need to have a Y Axis, although it is on by default. You can turn it off by unchecking the Y Axis box.

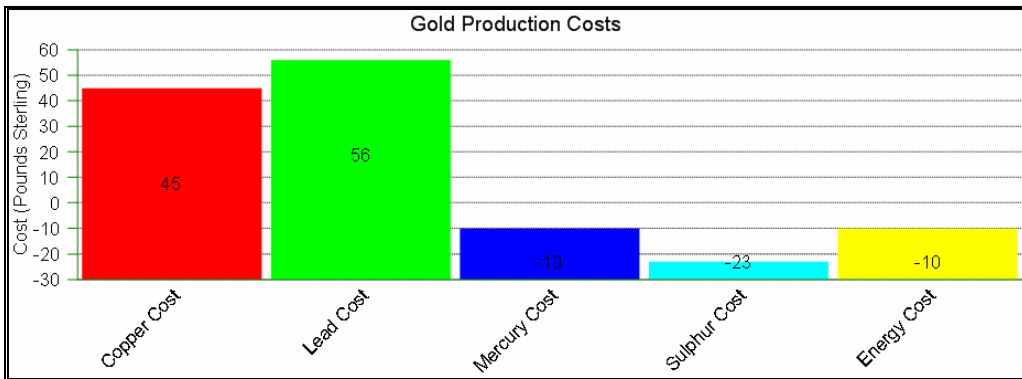
The scaling of the Y Axis can be either **Automatic** or **Range**. If Range is specified you must enter the lower and upper ranges for the axis, otherwise appropriate values will be calculated from the data.

The **Divisions** field specifies how many tick marks are shown, but it is not exact, Easy Report uses it as a guideline only and tweaks the actual amount to be a sensible number for the range of the Y Axis.

The **Baseline** dictates at what Y Axis value the bars are drawn from. Zero is the usual choice but can be anything that you want. Suppose we choose the value of 20 for the Baseline the output will be as shown below.

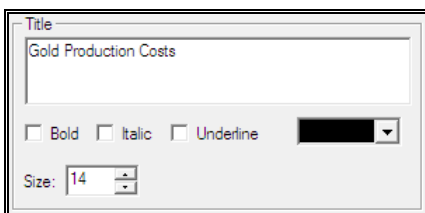


Selecting **Min** causes the baseline to automatically be placed at the minimum value in the data being trended (in our example here, -23, rounded to the nearest “nice” value).



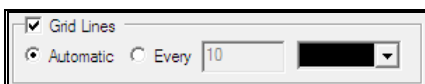
The **Label** is the text that appears alongside the Y Axis and this is displayed even if the Y Axis box is not checked. If you do not want to show any text, do not enter any here.

Title



The title appears at the top of the chart. Select the font style, size and colour.

Grid Lines



These are the horizontal lines on the examples shown above. If **Automatic** is selected the grid lines will be at the same positions as the tick marks on the Y Axis. If **Every** is selected then the gridlines will be at the intervals specified in the field. You can also set the colour of the grid lines.

Background Colour

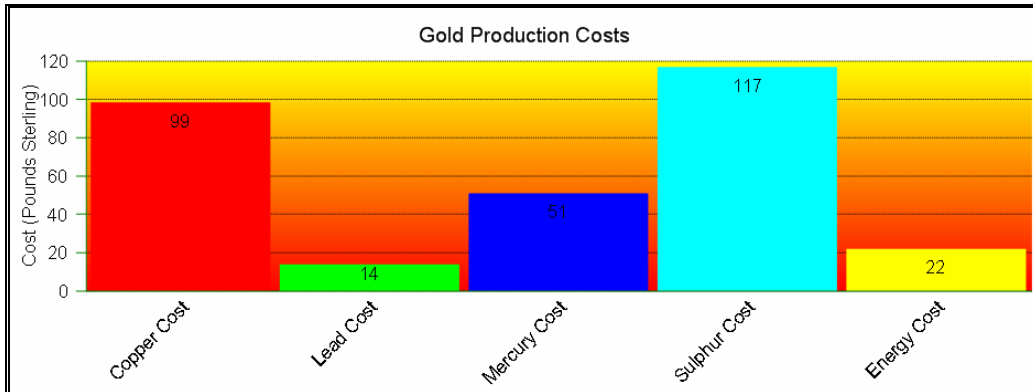


Background Colour

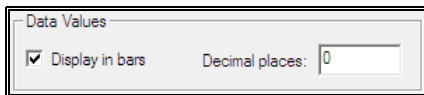
Transparent To

The background colour is the colour of the paper behind the bars. The default is for plain white, which is different from **Transparent**. If you check **Transparent** then anything *behind* the barchart (e.g. Field Values) will be visible through it.

You can do a graduated background by checking the **To** box and picking a second colour. The graduation always runs from the bottom of the chart to the top, ending with the **To** colour (see below).



Data Values

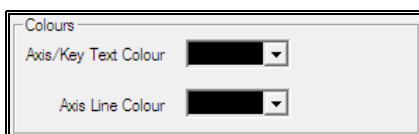


Data Values

Display in bars Decimal places:

You can display the actual data values for the bars inside the bars (these are shown on most of our examples here). If the **Display in bars** box is not checked, the numbers inside the bars are not shown. You can also specify the number of **Decimal places** to show.

Colours



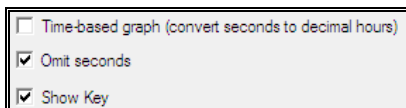
Colours

Axis/Key Text Colour

Axis Line Colour

You can change the colour of the text that appears on the axis and any keys for the information using this section; the Axis Line Colour can also be changed, as a default both colours are black.

Other Options

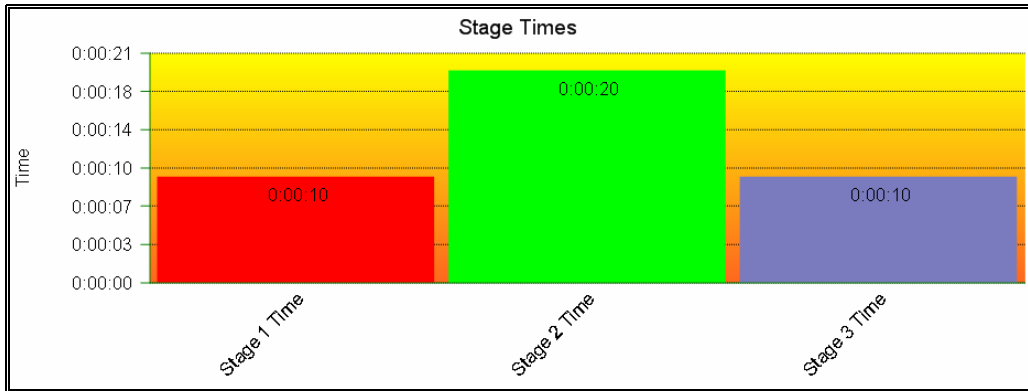


Time-based graph (convert seconds to decimal hours)

Omit seconds

Show Key

If you select **Time based graph** then the values of the barchart are assumed to be seconds and are converted to hours and minutes (displayed in the format (h)h:mm(:ss)). You can either include or exclude seconds by checking the **Omit seconds** box.



The **Show Key** box determines whether or not a key is shown at the top right of the barchart (a key is only relevant on series and stacked types of chart).

Bar Chart Items List

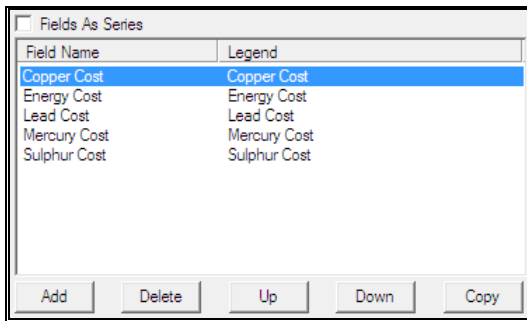
Field Name	Legend
Copper Cost	Copper Cost
Energy Cost	Energy Cost
Lead Cost	Lead Cost
Mercury Cost	Mercury Cost
Sulphur Cost	Sulphur Cost

This screen is where you define the data that is to appear in the chart. It's similar to defining rows on the Data Block item, described earlier.

Table

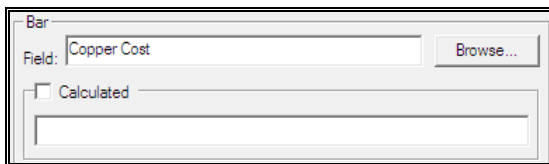
This is the table from which all of the data for this chart will be taken.

Field List



Use the **Add** and **Delete** buttons to add or remove items from the list. Use **Up** and **Down** to order the list (this is the order that the bars will be drawn, from left to right). Use **Copy** to copy an existing item. If you check the **Fields As Series** box the graph will swap what is shown in the Key with what is shown as the legend (X Axis) and vice versa. An example will be shown later under Other Styles Of Barchart.

Field Item

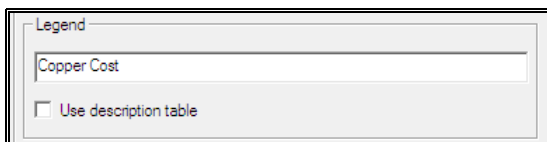


Use the **Browse** button to select a different field for this item, or type in a name if you are going to make a calculated field. Remember, if it *is* a calculated field the name must be unique in the current table.

To make a calculated field simply check the **Calculated** box and enter the expression into the field below. Remember that the expression must have field names in square brackets (e.g. [Total Cost] * [Total Amount]).

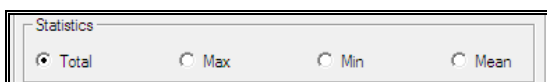
See the Data Block item for full information on calculated fields.

Legend



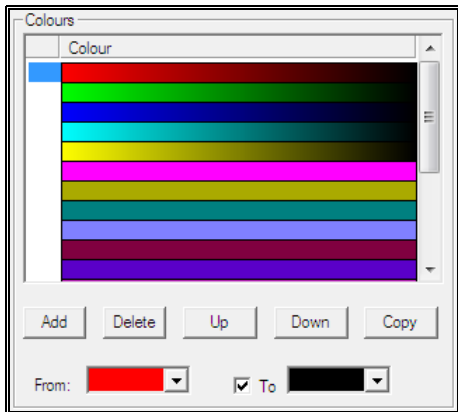
The legend is the label associated with this item. It defaults to the same name as the field when you **Browse** for a field, but you can change it to anything. If you check **Use description table** then the legend is looked up from a table in the database called *Descriptions*. This is specific to Downtime Monitoring applications and is not of general interest.

Statistics



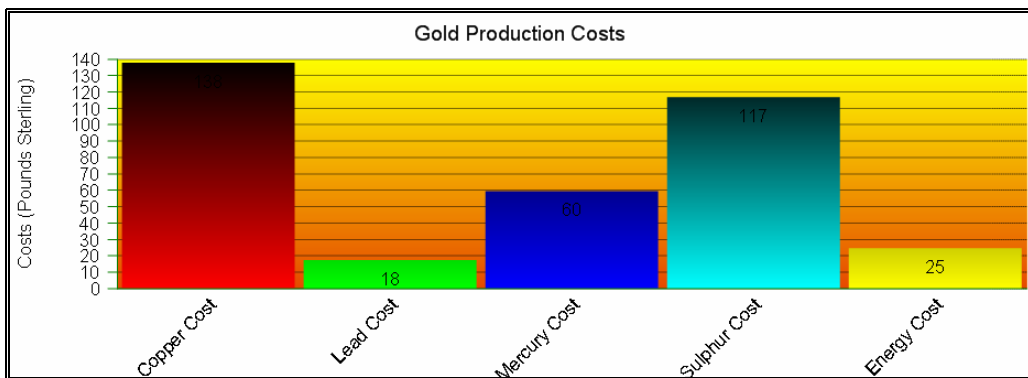
If the chart type is Statistical then you can select what type of statistic to use from the list shown in this area of the dialog. Make your selection by clicking the relevant radio button.

Colours



This is where you select the colours of the bars. The first bar is drawn in the first colour, the second bar in the second colour and so on. You can arrange the list however you like by moving colours **Up** and **Down** in the list and by clicking the **From** (and optionally **To**) colours to select new colours. You can also add new items to the list or remove unwanted items. Make sure there are enough different colours to cater for the amount of bars you might have.

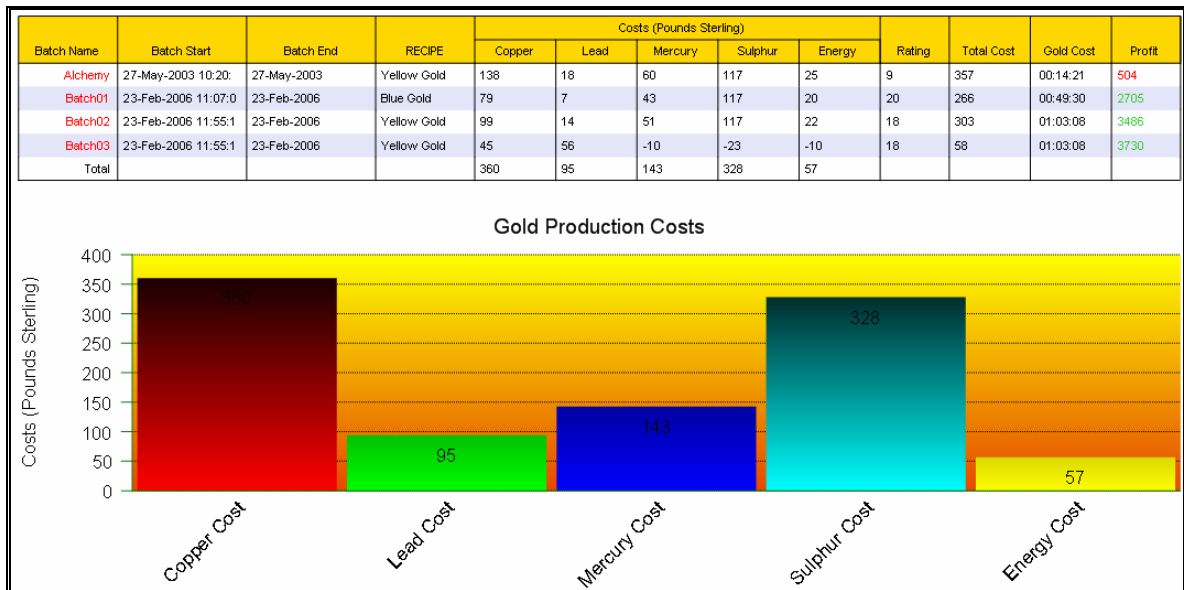
Using the **To** option produces some interesting results.



The shading is always across the total height of the chart, so items with a low value (like the 18 and 25 in the example above) will show minimal shading effects while larger values will show more. You have to be careful with colours, as always. In this example you can't see the values in the 1st, 3rd and 4th bars because of the shading colours.

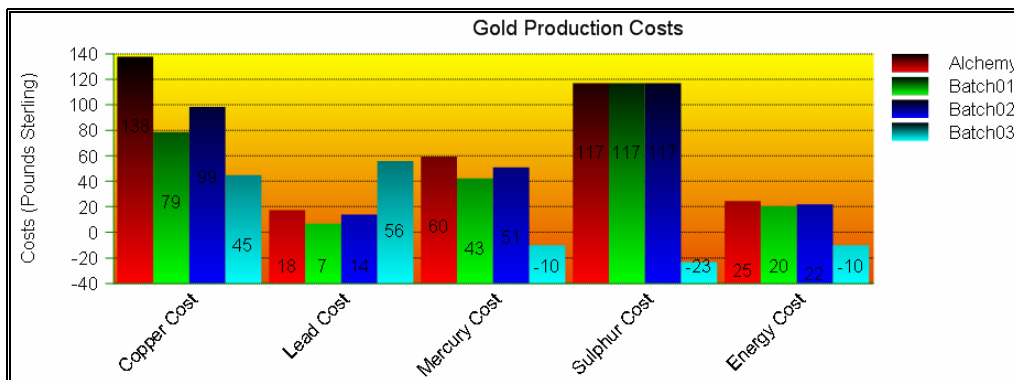
Other Styles Of Barchart

Statistical



A statistical barchart allows you to select more than one batch. Easy Report looks at the data in each batch and for each data item that you are reporting on calculates the statistic that you have requested (min, max, mean or total). The graph above is a graph of Total, and the data block items is shown too, for comparison.

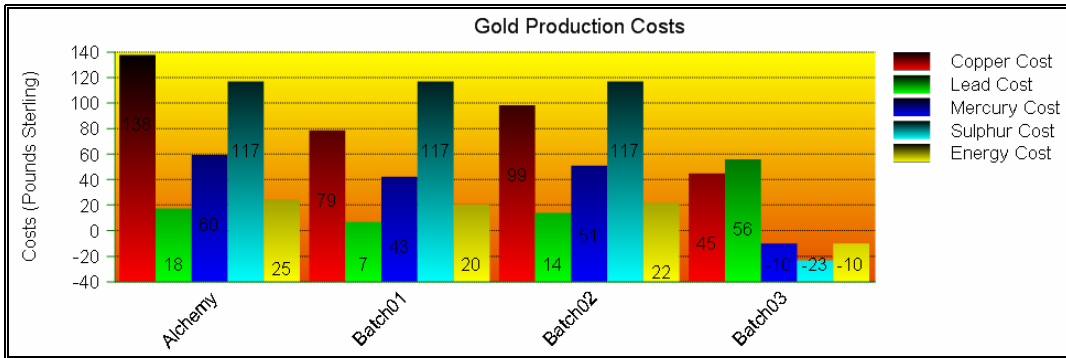
Series



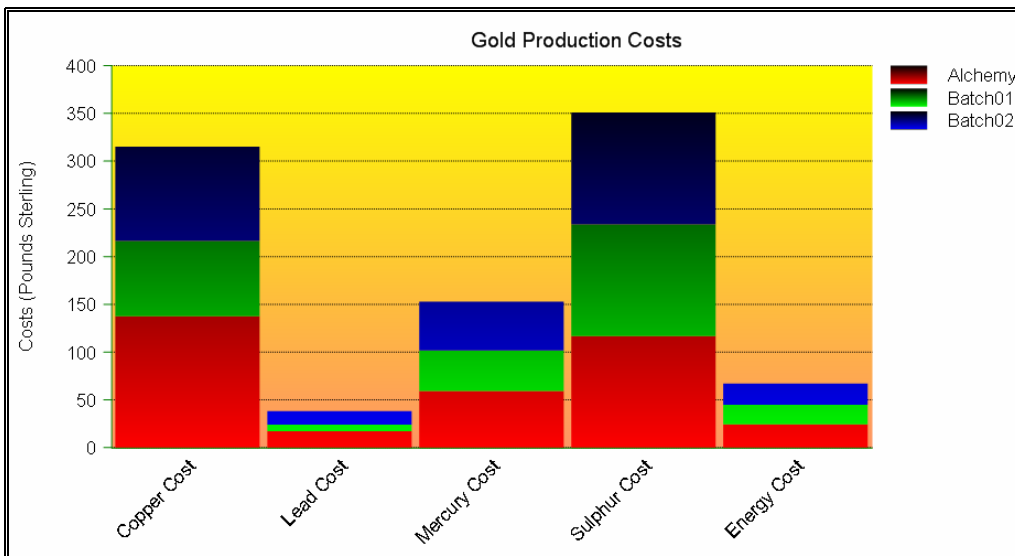
A series barchart shows the data from each batch alongside each other so that you can compare them. The key at the top right of the chart shows which colour refers to which batch.

You can see from this example that the sulphur costs in three of the batches were identical, but totally different in Batch03.

If you had checked the Fields As Series box in the configuration, the Key and the legend (X Axis) would be swapped, as shown below.

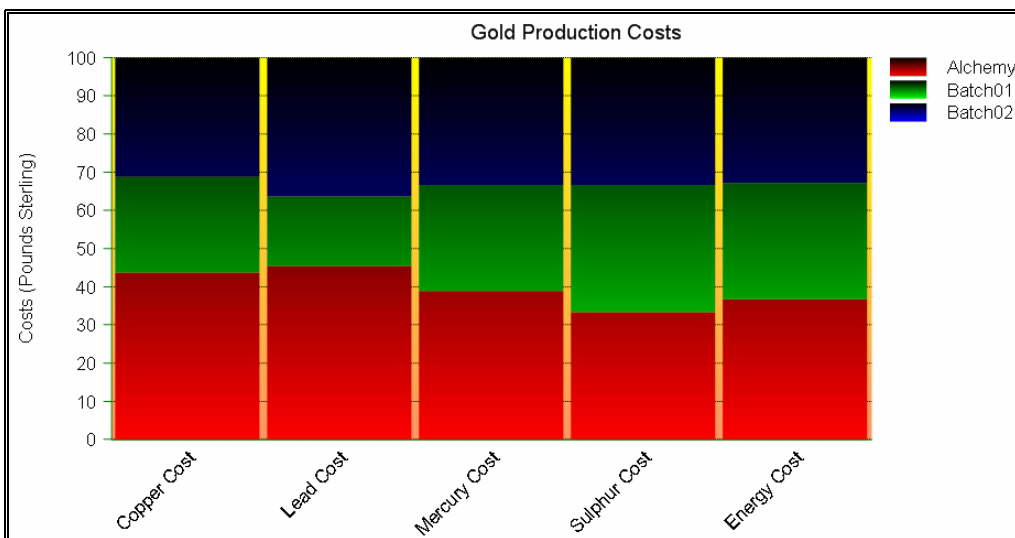


Stacked



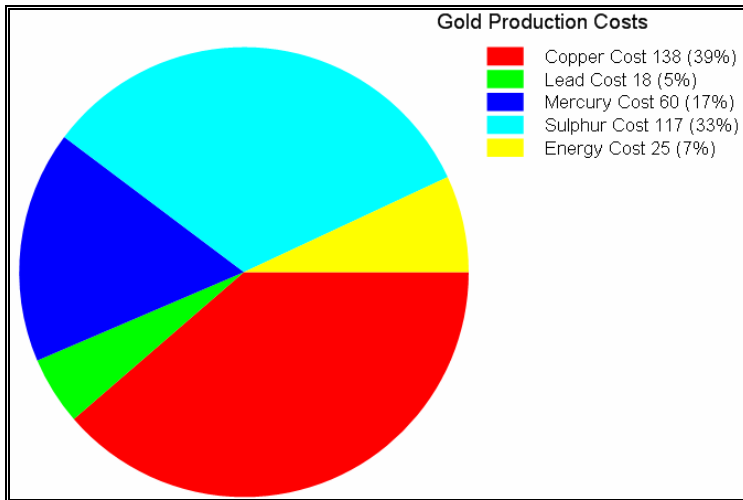
Stacked barcharts show the corresponding data items from each batch stacked on top of each other. Again, the key shows which colour refers to which batch.

Stacked 100%



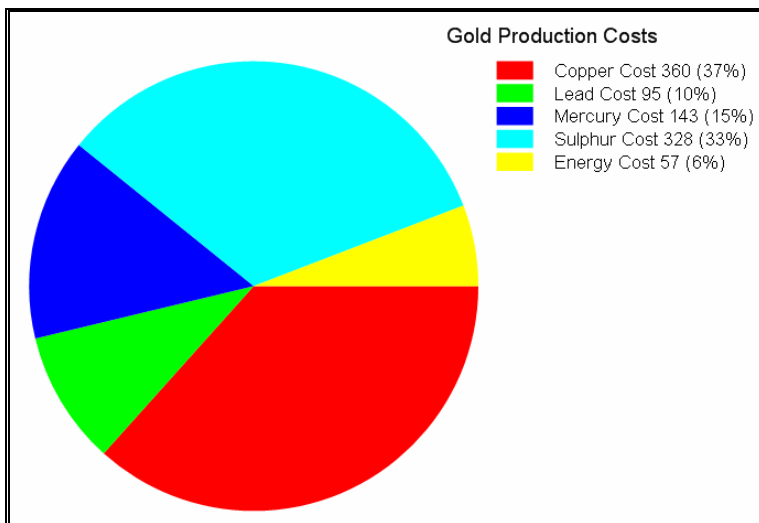
The stacked 100% chart is similar to the stacked chart except that all of the stacks are drawn to the same height (100%) and the data that forms each stack is displayed as a percentage of the total.

Simple Pie



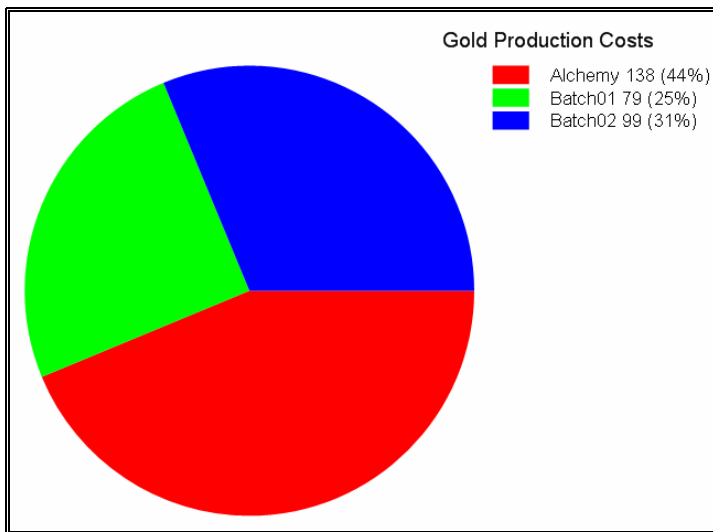
Simple pies accept data from only one batch and show the breakdown of each data item as a percentage of the total.

Statistical Pie



The statistical pie takes data from each batch that you select and evaluates the relevant statistic (in this case Total). A pie is then constructed from this statistical data, again showing each data item as a percentage of the total pie.

Series Pie

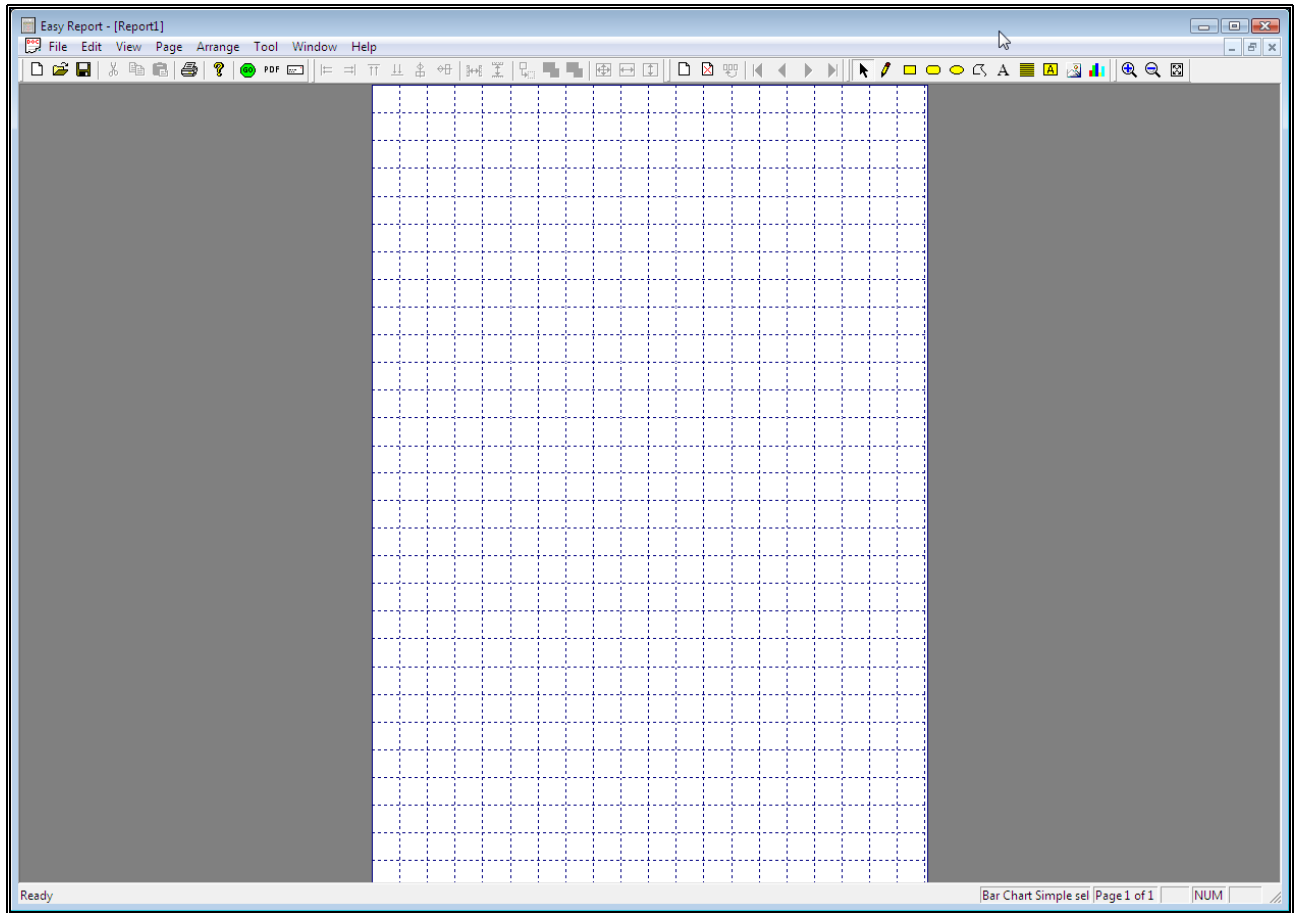


The series pie totals up the data items selected for each batch and displays a pie made up of these totals, so that the pie has a segment for each batch selected. Using the example data that we have been using throughout this section, each segment is the total of the individual cost items on a batch-by-batch basis.

Getting Started With Easy Report

This guide will show you how to get started with Easy Report. In it, we will use the Alchemy template to produce a simple report using most of the tools available in Easy Report.

Create a blank report

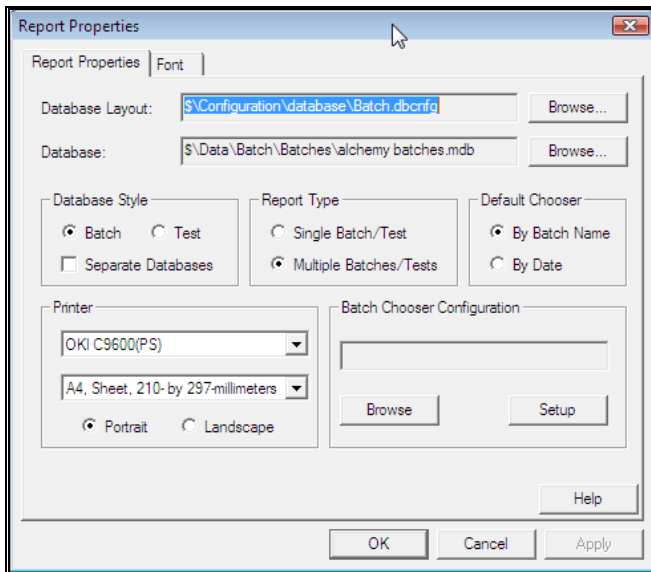


Select **File->New**, maximise the document window and click Zoom To Fit.



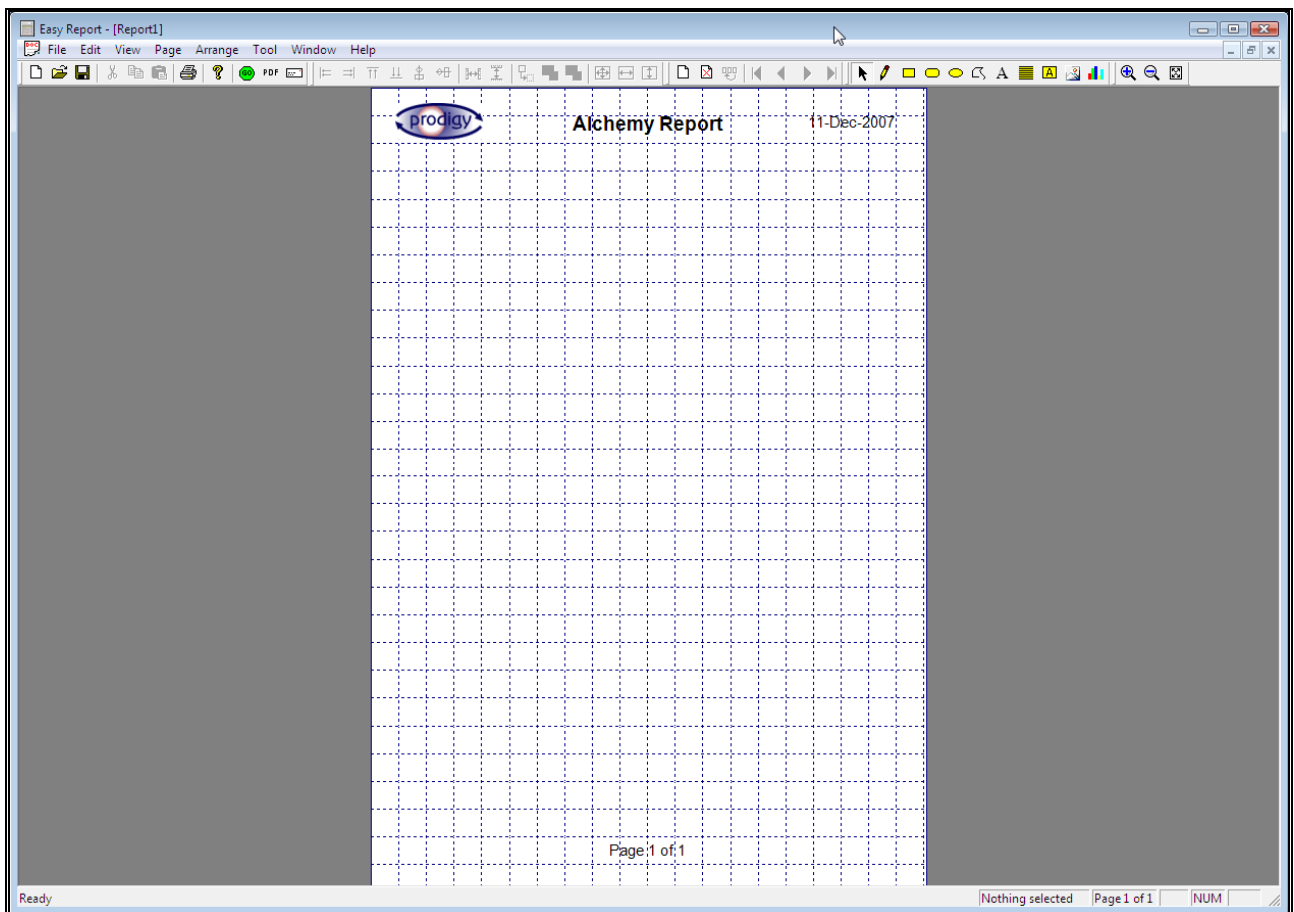
You will now have an empty report taking up the whole display.

Set the report properties



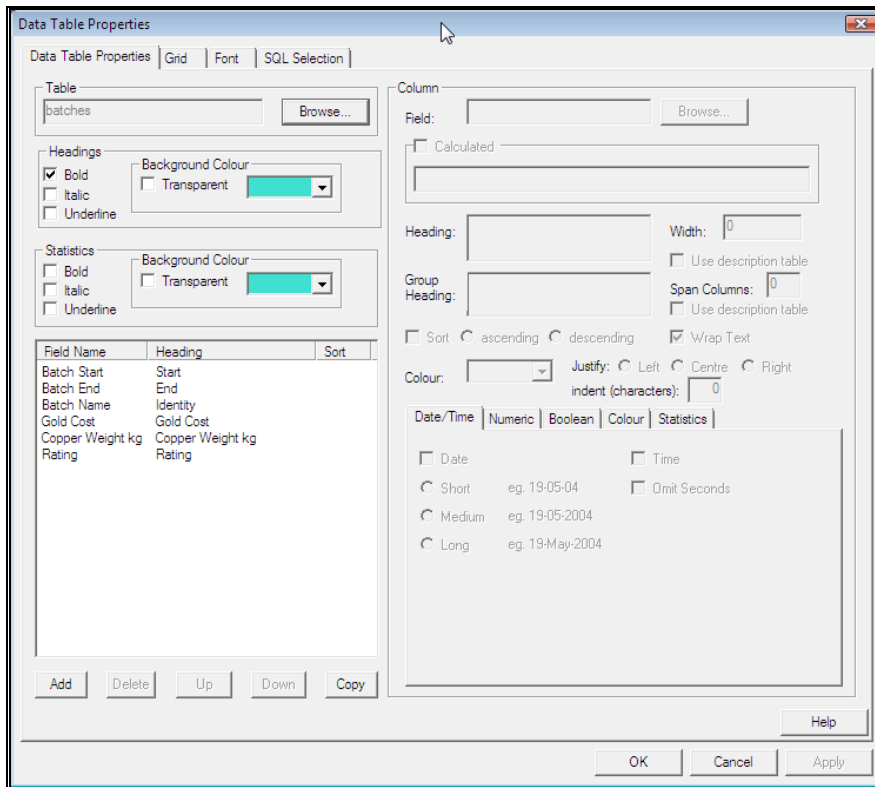
Select **Edit->Report Properties**. Set the *Database Layout* to *Batch*, the *Database* to *Alchemy Batches.mdb*, *Report Type* to *Multiple Batches/Tests* and the *Default Chooser* to *By Date*.

Place headings and footers



Use the text and bitmap tools to display a logo, report heading, the date and time the report is printed, and the page number.

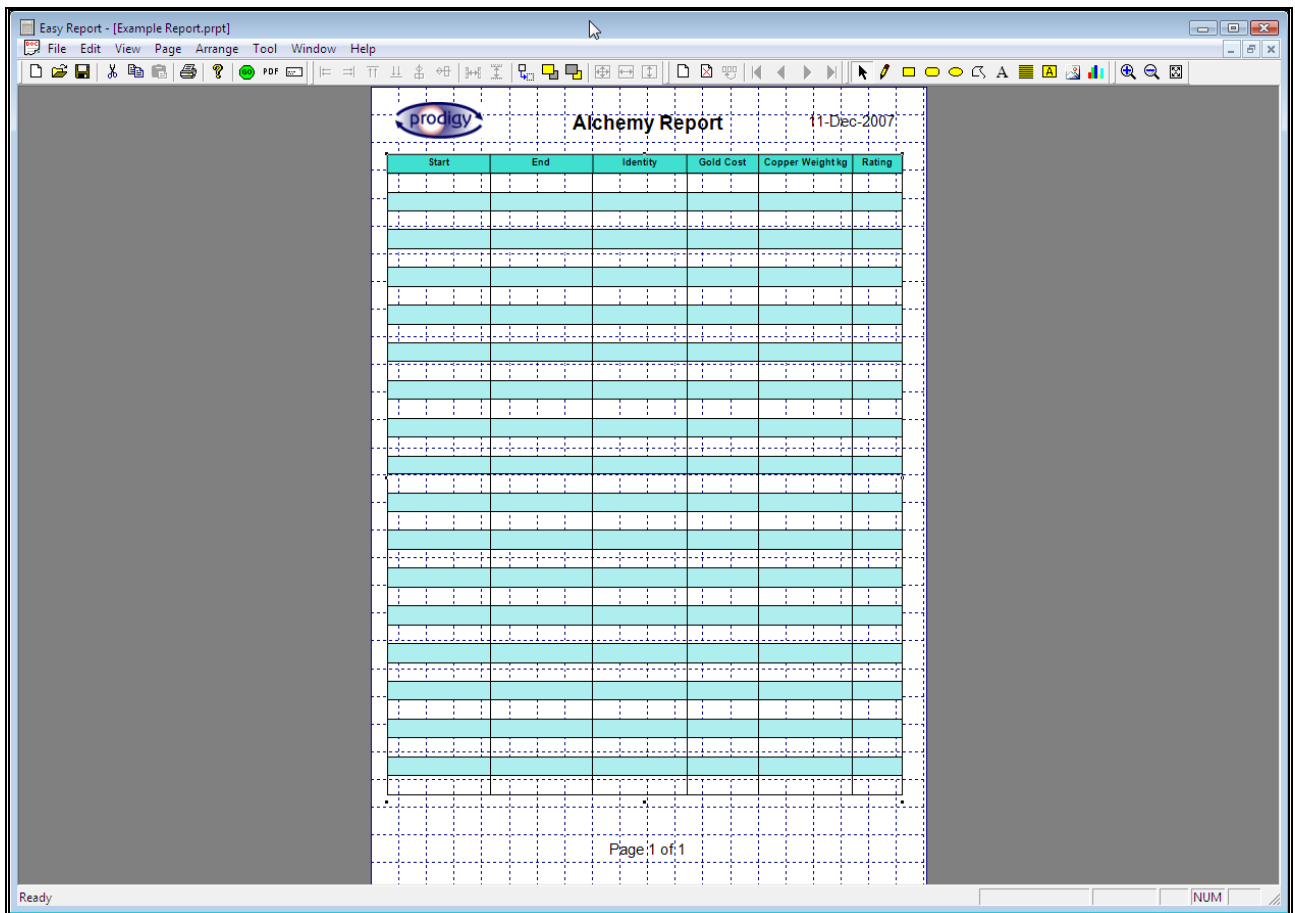
Place a data block



The data block is the main part of our report. Drag it out to take up most of the height of the blank area in the middle of our report. The width will resize to accommodate the number of columns we add.

- The fields **Batch Start**, **Batch End**, **Batch Name** and **Rating** are all pre-defined in the batch database, so they can be picked using the **Browse** button.
- The **RECIPE** and **Gold Weight (kg)** fields will need to be manually typed into the field text box, as they are not pre-defined in the database.
- Statistics have also been added for the **Gold Weight (kg)** and **Rating** fields.

The page will look like this:



When you run the report you get something that looks like this:-

The screenshot shows a window titled "Easy Report - [Example Report.prpt]" with a menu bar containing "Print", "Next Page", "Prev Page", "Two Page", "Zoom In", "Zoom Out", and "Close". The report content includes the Prodigy logo, the title "Alchemy Report", and the date "11-Dec-2007".

Start	End	Identity	Gold Cost	Copper Weight kg	Rating
20-08-2007 16:36:21	20-08-2007 16:39:19	20070820TEST	4305	64	24
22-08-2007 08:55:54	22-08-2007 08:59:07	20070822TEST	4305	47	22
23-08-2007 15:29:14	23-08-2007 15:32:17	20070823TEST	4305	46	10
10-09-2007 15:31:18	10-09-2007 15:34:07	20070910TEST	3874	46	18
10-09-2007 17:15:40	10-09-2007 17:22:53	20070910TEST2	2193	72	8
11-09-2007 11:49:58	11-09-2007 11:53:14	20070911TEST	3014	28	14
12-09-2007 16:24:02	12-09-2007 16:27:29	20070912TEST1	3444	51	13
12-09-2007 16:28:58	12-09-2007 16:32:09	20070912TEST2	1722	72	10
04-10-2007 13:53:44	04-10-2007 14:04:36	20071004TEST1	4305	46	9
08-10-2007 12:13:57	08-10-2007 12:17:16	20071008TEST1	3874	28	24
08-10-2007 14:14:16	08-10-2007 14:17:20	20071008TEST2	4305	46	15
08-10-2007 16:43:55	08-10-2007 16:47:06	20071008TEST3	3874	45	8
11-10-2007 12:33:48	11-10-2007 12:36:53	20071011TEST1	4305	46	17
11-10-2007 16:37:48	11-10-2007 16:40:51	20071011TEST2	4305	45	5
15-10-2007 09:36:10	15-10-2007 09:38:56	20071015TEST1	4305	46	13
15-10-2007 09:42:51	15-10-2007 09:46:01	20071015TEST2	2307	64	1
15-10-2007 09:47:26	15-10-2007 09:50:33	20071015TEST3	2307	65	19
15-11-2007 10:23:57	15-11-2007 10:27:01	20071115TEST1	4305	64	9
20-11-2007 17:08:05	20-11-2007 17:11:19	20071120TEST1	3444	45	9
07-12-2007 09:10:21	07-12-2007 09:13:24	20071207TEST1	3444	64	9
Total			72203	1030	
Mean			3510	52	13
Max			4305	72	24
Min			1722	28	1

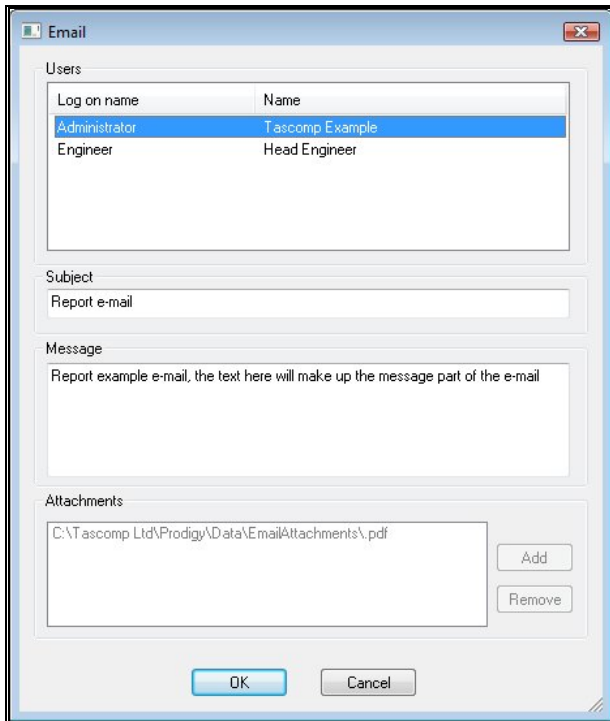
Page 1 of 1

Page 1

NUM

Email Reports

Reports (both Easy Reports and Proof of Process Reports) that are created in Prodigy can be automatically sent as a PDF attachment in an Email. Pressing the Email buttons on the main toolbar will load up the following dialog that allows an Email to be created.



In order to use this facility Prodigy's **Email Manager** must be set up and also **Users** must be configured to have an Email address associated with them. Details of how to configure both Email manager and users can be found in the Alarms and Users sections of the manual.

To send an Email select the user(s) that it needs to be sent to (use ctrl-click to select multiple entries from the Users list), enter the Subject and Message of the Email and click ok. This will then run the report, produce a PDF of the report and then Email it as an attachment to the users that have been selected. The report attachments are selected automatically when an Email is sent using one of Prodigy's reporting tools.

The sending of an Email can be automated so that the above dialog does not appear by specifying the appropriate parameters to the program when a report is run. These parameters can provide all of the information supplied by the dialog. The list of parameters that can be used is specified in the appendix to this manual.

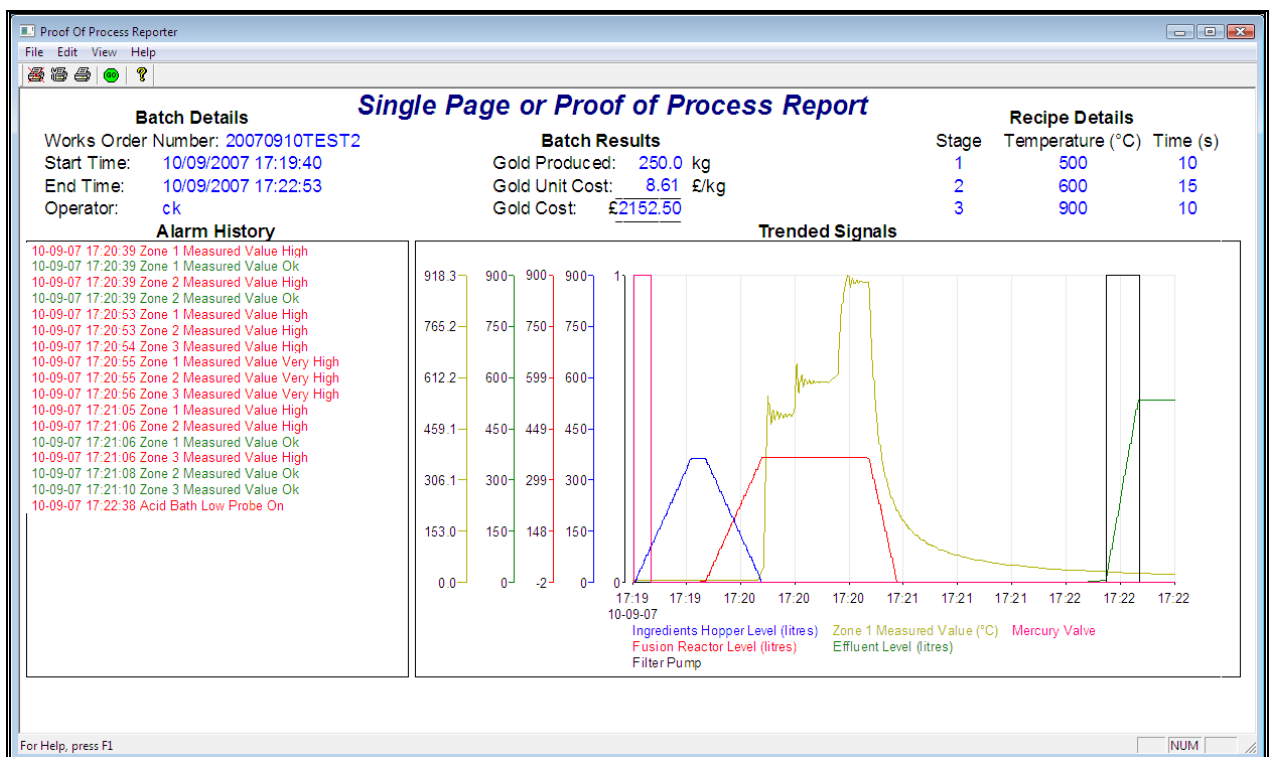
Proof Of Process Report

Proof Of Process Reports

Proof of Process Report allows you to produce, on one page, a report detailing all of the key items of a batch run. These are:

- Data fields from the Batch Database.
- Trends of signal data recorded over the duration of the batch run.
- Alarm records from the Batch Database.

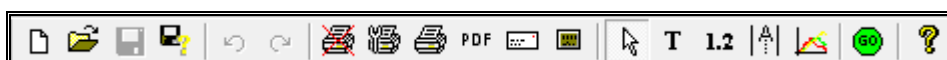
The Proof of Process Report dialog is shown below.



The Proof Of Process Reports are not normally run in a window, as shown above, this is mainly for design use. The Proof Of Process Reports are usually run by BDR when a batch ends and are automatically sent to a printer.

Configuring A Proof Of Process Report

Configuring a Proof Of Process Report is very much like configuring a Display. You select the type of object you want to place onto the report, drag it to the correct size and position, assign properties to it, align it with other objects, etc. and save your finished report into a document that can be retrieved and modified later. Again like Display Builder, the Proof Of Process Report program makes use of toolbars to control the functions provided by the program.
















The facilities provided by the main toolbar (reading from left to right) are as follows:

- **New.**



Begins a new report configuration. The existing configuration will be closed and if any changes have been made and not saved, you will be notified.

- **Open.**
 Open an existing configuration. A browser window will be displayed allowing you to choose the existing report to be opened.
- **Save.**
 Save the changes that have been made to the existing configuration. This button is only available if changes have been made, and so serves as a visual reminder as to whether or not changes have been made.
- **Save As.**
 Save the existing configuration under a name to be specified. A standard browser window is displayed that allows you to enter the name for this configuration.
- **Undo.**
 The Undo button can be used to reverse the last operation and/or remove the last entry that was typed in.
- **Redo.**
 The Redo button can be used to reverse the last “undo” operation that was performed.
- **Clear Printer.**
 Clears any printer configuration information that is present. By clicking this button there will be no printer selection or configuration information attached to this report and the program will prompt for a printer when it runs. Useful mainly when the printer that was selected and saved with the report no longer exists.
- **Printer Configuration.**
 Displays a standard printer configuration dialog. Use this dialog to select the printer to which output from this report is to be sent. Use the Properties button on this dialog to set any properties required (e.g. Paper orientation, resolution). This information is stored with the report. When the report is run by BDR it will be to the printer specified here and with the properties specified for that printer. If no printer is configured for a report, it will go to the default printer with default properties.
- **Print.**
 This prints the current report.
- **PDF.**
 Outputs the current report to a PDF file (requires PDF output license option)..
- **Email.**
 Sends the current report via email (requires the license option for emailing). To see how to send an Email please refer to the **Report Email** within the **Easy Report** section of the manual.
- **Properties.**
 This allows you to set various attributes about the report.
- **Select.**
 Chooses the Select tool, a pointer used to select objects that have already been placed on the report in order to position or resize them.
- **Text.**
 Selects a Text object as the next item to be placed on the report.

- **Database Field.**



Selects a Database Field object as the next item to be placed on the report.

- **Alarm List.**



Selects an Alarm List object as the next item to be placed on the report. Only one alarm list may be placed on a report.

- **Trend.**



Selects a Trend object as the next item to be placed on the report.

- **Go.**



Runs the current Proof Of Process Report. A Batch Chooser dialog will be displayed in order for you to select a batch.

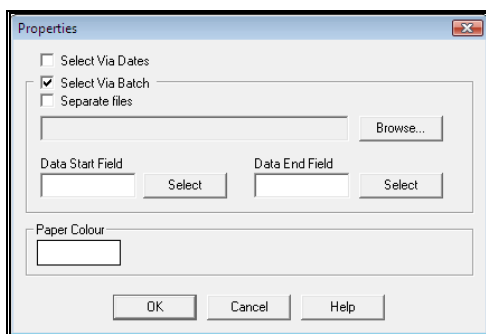
- **Help.**



Invokes the Help system.

Select A Database

Begin by clicking the *Properties* button on the main toolbar, as described above. This displays the dialog shown here.



- If you want to select when data is taken for the proof of process report via a date dialog that pops up when the report is run then check this box.
- If you are working with separate database files per batch, check the *Separate files* box, otherwise leave it unchecked.
- Use the *Browse* button to select either the database file or the directory in which the database files reside, depending on whether or not the Separate files box is checked.
- The **Data Start** and **Data End** Fields can allow the system to provide an alternative starting or end point for data that is viewed in a Proof of Process Report. For example if a process takes 30minutes to warm up you may not be interested in this 30minute period; therefore you would store the time and date when the portion of the batch you were interested in started and read this in as a value for when to start showing data on the Proof of Process Report

Placing Objects

Select the object type you want to place by clicking the relevant button on the main toolbar, Text, Database Field, Alarm List or Trend. Move the mouse pointer to roughly where you want the object to be placed and click. The object will be positioned with its top-left corner at the location of the mouse pointer. If snap-to-grid is turned on, the top left corner of the object will snap to the top left corner of the grid square in which the mouse is clicked.

Continue to place objects as required and then use the Select tool to move or resize any objects as required.












Objects may be aligned and sized relative to each other in much the same way as they can in Display Builder. To do this select two or more objects and then use one of the functions on the **Alignment Toolbar**, shown below.









To select multiple objects, click on the first object you want to select, then hold down the **shift** key as you click on other objects. They will all become selected. If you click on an object that is already selected while you are holding down the shift key, it will become unselected. Alternatively do a *marquee selection* by dragging the mouse over the entire area in which you wish to select objects. A marquee box will be drawn as you drag.

The last object that you select is known as **the dominant control**. When you align objects they are aligned against the dominant control, i.e. the dominant control does not move, all of the other controls do.

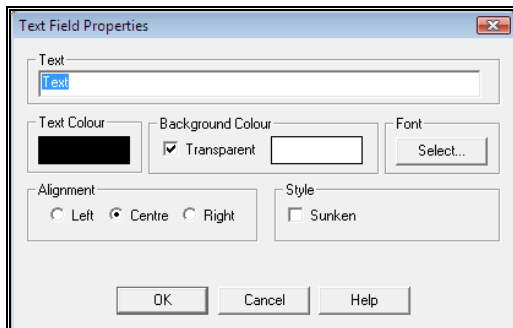
The functions of the Alignment Toolbar are as follows (reading from left to right)

- **Align Left.**
 Lines up the left-hand edge of each selected item with the left hand edge of the dominant control.
- **Align Right.**
 Lines up the right hand edge of each selected item with the right hand edge of the dominant control.
- **Align Top.**
 Lines up the top edge of each selected item with the top edge of the dominant control.
- **Align Bottom.**
 Lines up the bottom edge of each selected item with the bottom edge of the dominant control.
- **Align Centre (horizontally).**
 Lines up the horizontal centres of each selected item with the centre of the dominant control.
- **Align Centre (vertically).**
 Lines up the vertical centres of each selected item with the centre of the dominant control.
- **Show Grid.**
 Displays the grid, drawn at the selected granularity.
- **Snap To Grid.**
 Causes the top left hand corner of the object to be snapped to the top left hand corner of the grid square in which the mouse is clicked.
- **Set Grid Granularity.**
 Displays a dialog that allows you to specify the granularity of the grid.
- **Distribute Horizontally.**
 This button is only available when you have three or more objects selected. The system finds the left most edge of all of the objects and the right most edge. The objects that have the left and right most edges are left unmoved, while all other objects are spaced evenly between the outer two edges.
- **Distribute Vertically.**
 This button is only available when you have three or more objects selected. The system finds the top most edge of all of the objects and the bottom most edge. The objects that have the top and bottom most edges are left unmoved, while all other objects are spaced evenly between the outer two edges.

- **Make The Same Size.**
 All objects are made the same size as the dominant control.
- **Make The Same Width.**
 All objects are made the same width as the dominant control.
- **Make The Same Height.**
 All objects are made the same height as the dominant control.
- **Duplicate.**
 Makes a copy of the selected control. All attributes that have been set for that control will be copied too.
- **Group.**
 If multiple objects are selected then pressing this button “groups” them together, they can then be moved as a group and certain properties edited on all objects in the group. Individual objects within a group can be selected and their individual properties edited by holding down the control key and clicking on the required object.
- **Ungroup.**
 If multiple objects have been “grouped” then pressing this button will ungroup them so that they are all individual objects again.

Object Properties

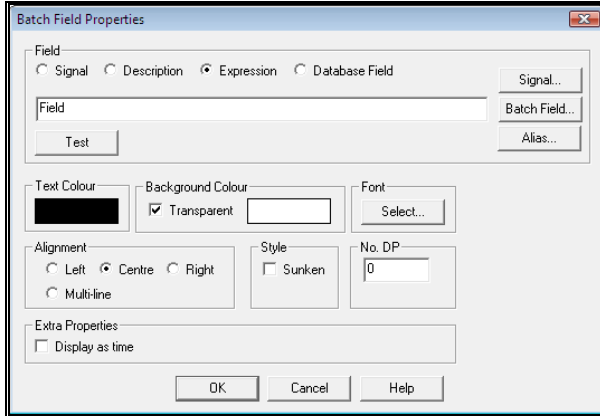
Text Object



- **Text.**
Enter the text you want to appear in this field.
- **Text Colour.**
Click on the coloured button to display the standard colour chooser dialog. This will be the colour of the text itself.
- **Background Colour.**
Click on the coloured button to display the standard colour chooser dialog. This will be the colour of the box the text sits in. If you check **Transparent**, the text box will have no colour, but will allow the colour of the paper to show through.
- **Font.**
Click this button to show the standard font selector dialog. Select the required font for this item.
- **Alignment.**
Click the relevant button depending on whether you want the text to be **left**, **centre** or **right** justified within the box you have defined for it.
- **Style.**
Check the **Sunken** box if you want the text to appear as if it is inside a sunken box, such as in the example shown below.

Batch01

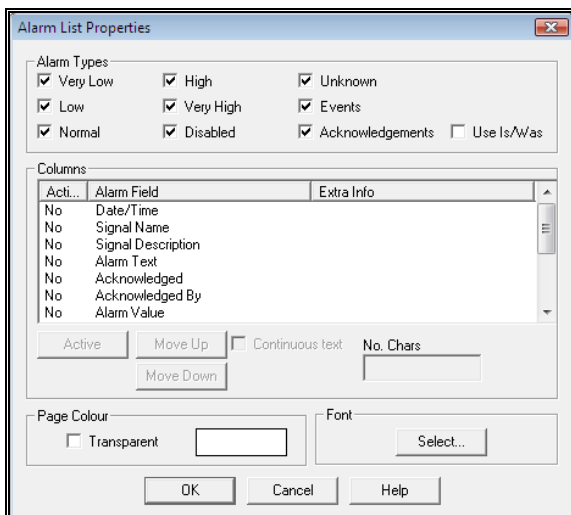
Database Field



The properties for this type of object are the same as that of the **Text** object, but with the following additions:

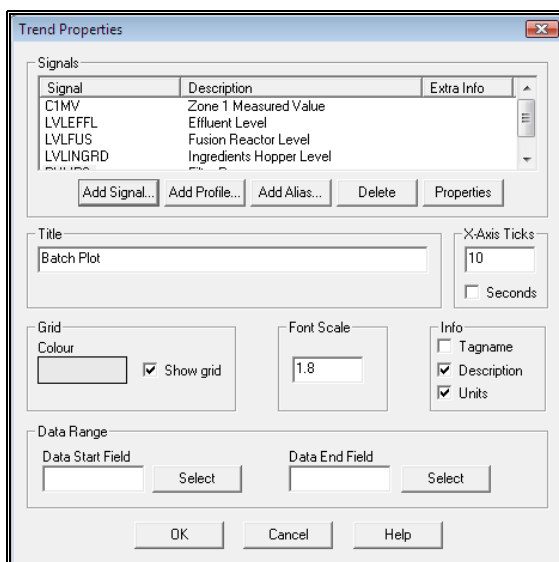
- **Field.**
The four radio buttons allow you to select a **Signal**, **Description**, **Expression** or **Database Field** to be written into the field; which radio button is selected will vary the number of options available. The signal radio button will display the value of a signal that is selected, the description will display the description of a particular signal and an expression can combine database fields and signals to perform some calculation. Database Field will obtain data from a database that has been selected. Use the **Batch Field** button to display a list of all fields within the database (the database name must have been given and the database must exist for this option to be available). If no database has been given yet, or the required field does not yet exist in the database, you can enter the name of the field manually.
- **Multi-line.**
Allows a string to be split over multiple lines. Use this option if the field you want to display is quite long (a string, usually). You should size your field so that it is deep enough to display multiple lines.
- **No. D.P.**
Enter the number of decimal places to which numeric data should be displayed.
- **Display as time.**
Display the value as a time field.

Alarm List

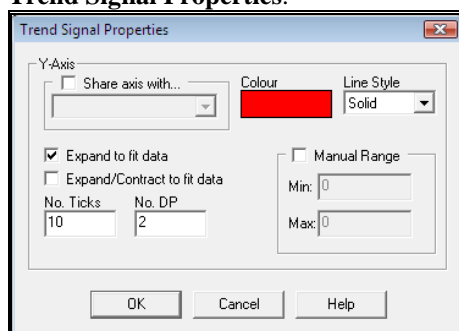


- **Alarm Types.**
Check each of the alarm types you want to appear in your alarm list.
- **Use Is/Was.**
If this is checked the alarm text will read something like “Acid flow rate was low”, if it is not then it will read like “Acid flow rate low”.
- **Columns.**
Make each column that you want to appear in your alarm display **Active**.
Use the **Move Up** and **Move Down** buttons to adjust the order in which these columns appear.
Each new column begins after the longest entry in the previous column, so each column should line up properly. If you check the **Continuous text** box for a given column, it will cause this column to be treated as a continuation of the previous one, i.e. as if it were one column. The **next** column will begin after the longest **combined** entry formed from two or more continuous text entries.
Use the **No. Chars** field to specify how many characters of the given column to display. If this value is left blank, all characters will be displayed.
- **Page Colour.**
Sets the background colour of the box in which the alarm list is displayed.
- **Font.**
Select the font required for the alarm list text.

Trend



- **Signals.**
Add the **signals** and/or **alias tags** that you want to display in your trend window by pressing the relevant button. Delete signals or alias tags by selecting the required one from the list and pressing the **Delete** button. Set any properties for a given signal or alias tag by selecting it from the list and pressing the **Properties** button.
- **Trend Signal Properties.**



- **Share axis with.**
If the currently selected trend has a y-axis that has the same y-axis range as another trend you can save space on the trend by sharing the axis. Check this box and choose the trend with which you want to share from the drop down list.
- **Colour.**
Select the colour for this trend. The graph and it's axis will both be drawn in this colour.
- **Line Style.**
Select the line style to use for this trace.
- **Expand to fit data.**
If this box is checked the y-axis will be expanded (i.e. the range will be increased) in order to fit the minimum and maximum data values of this graph. If neither this nor the next check box are selected the y-axis will have a range equal to the signals range.
- **Expand/Contract to fit data.**
If this box is checked the y-axis will be scaled so that it exactly fits the data for this graph. If the signals range is greater than that of the data, the y-axis will be contracted, if the data is greater than the signal range the y-axis will be expanded.
- **No. Ticks.**
Specify the number of tick marks to be drawn on the y-axis.
- **No DP.**
Specify the number of decimal places to which the labels on the y-axis will be drawn.
- **Manual Range.**
If you want to override the range of the signal check this box and enter min and max range values.
- **Title.**
Enter the title of your batch plot. This will be drawn at the top of the plot. Leave blank if you do not want a title.
- **X-axis ticks.**
Specify the number of tick marks you want on the x-axis.
- **Seconds.**
Check this box if you want the times on the x-axis to go down to seconds, otherwise they will only go down to minutes.
- **Grid Colour.**
Select the colour for the grid that is drawn on the trend object to aid data read-off.
- **Show Grid.**
Check this box if you want to display the read-off grid on the trend object, otherwise leave it blank.
- **Font Scale.**
Specify a relative scale for the text on the trend. Select a number larger than 1 to increase the size over the default and less than one to decrease it.
- **Info (Tagname, Description & Units).**
Specify which signal information you want to be displayed with the trend. You can select any or all of **Tagname**, **Units** and **Description**. If the batch was recorded by **BDR** with **Signal Attributes** selected for some or all of the recorded signals, these recorded attributes will be used in place of the ones currently in place for the signal (see **BDR** for more information).
- **Data Range.**
If there are sections at the start or end of a batch that you are not interested in been on the trend the **Data Start Field** and **Data End Field** can be used to select a different start and/or end point for the timescale. They can also be useful if you have a very long batch time with many stages to single out specific stages within the batch.

Running A Proof Of Process Report

Proof Of Process Reports may be run in one of two ways, either from BDR or from a button on either the **Menu Bar** or on a **Display Builder** Display. See the section on **BDR Configuration** for more information on the former and the sections on **Display Builder (Button Properties)**, **Menu Bar Configuration** and **run-time parameters** for more information on the latter.

Alternatively, the Proof Of Process Report can be run and printed from the configuration program, using the **Go** and **Print** buttons.

Multigraph

X-Y Plots

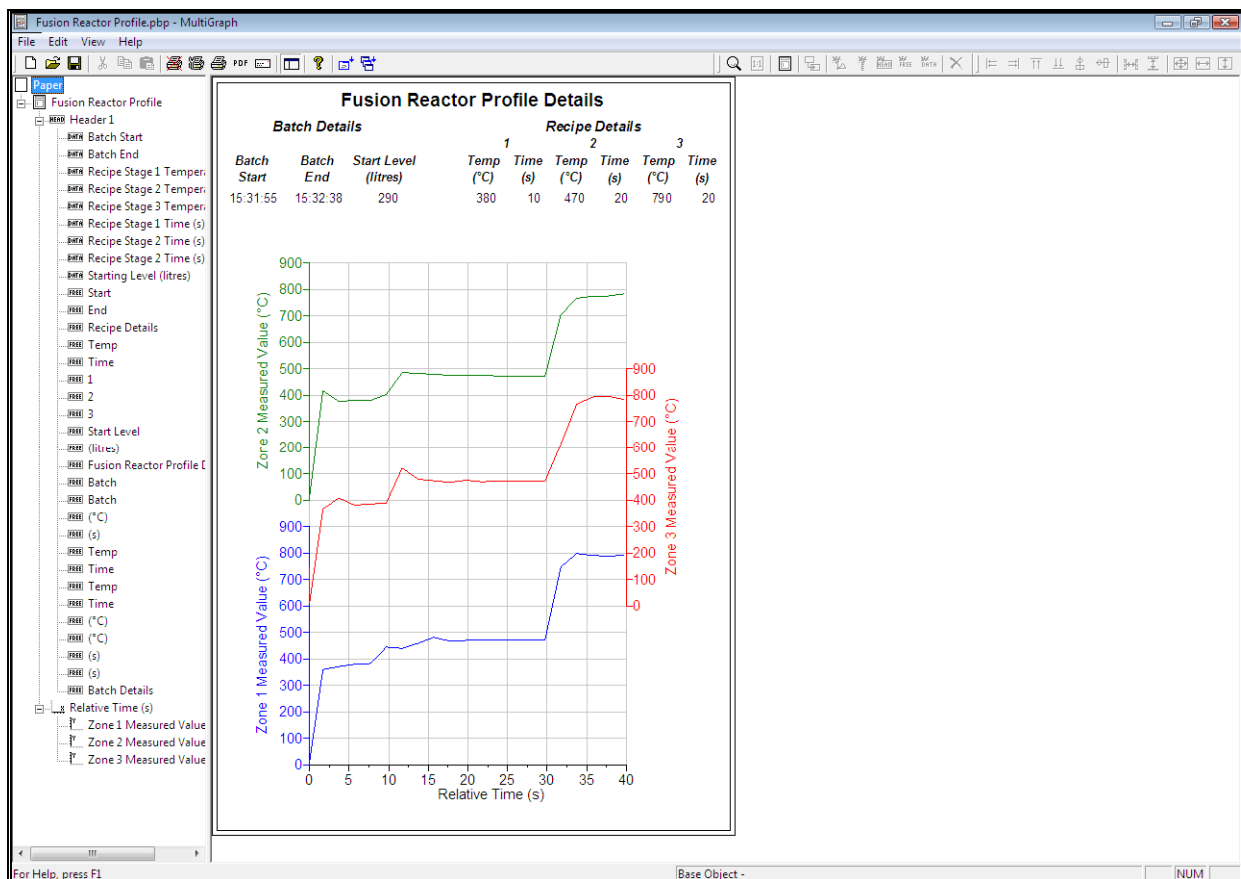
One type of graph that is often used is an X-Y plot. This simply plots any Y variable against any other X variable (provided there is some sort of common relationship between the X and the Y values). Multigraph provides facilities to plot any number of Y values against a related X value and to overlay many different data runs on the same plot, for comparison purposes. It also provides the facility to plot different sets of X-Y mappings on to separate graphs on the same plot, and also allows for batch specific data to be displayed in one or more graph headers.

Database Considerations

Each record in the database must be a “snapshot” of the system at a given time, by which we mean that all variables in the record must be recorded at the same time and preferably the time of the recording should be one of the fields.

Using Multigraph

The Multigraph program is shown below



As you can see it is split into two panes – the left-hand pane is a hierarchical list of all of the components of the plot and the right hand pane is the plot itself. At the top is a series of toolbars that give access to all of the main functionality of the system, and a menu bar that gives further options.

The Multigraph graph is hierarchical.

- The **Paper** is at the top level.
- The **Plot** is next and can be sized and positioned to be anywhere on the paper. A plot automatically has an X-axis.
 - **Y-axes** can be positioned on either side of the X-axis , and any number of Y-axes can be added to an X-axis.
 - **Fixed text** items can be added to a plot. These can be used for any extra labelling that is required, and they can be positioned anywhere within the graph area.
 - **Headers** may be added to a plot and this is an area for displaying non-graphical information. Each header can be positioned anywhere within the plot area.
 - **Fixed text** items can be added to a header. These can be used for any labelling purposes, and they can be positioned anywhere within the graph area.
 - **Data text** items can be added to a header and these are used to display non-graphical data from the database. They can be positioned anywhere within the header.

When any item is moved, all of the sub-items attached to it move with it, maintaining their position relative to each other and to the object being moved.

If you delete an object, all sub-objects are also deleted. **Beware** – there is currently no undo in Multigraph.

Some properties (**database table** and **font**) can be set for an object and they will propagate down to all sub-objects that do not have this property explicitly set. For example, if you change the font in the header object, all sub-objects (fixed text and data items) will use that same font. It's a way of maintaining uniformity and also of minimising the amount of changes you have to make if you want to change the font, for example.

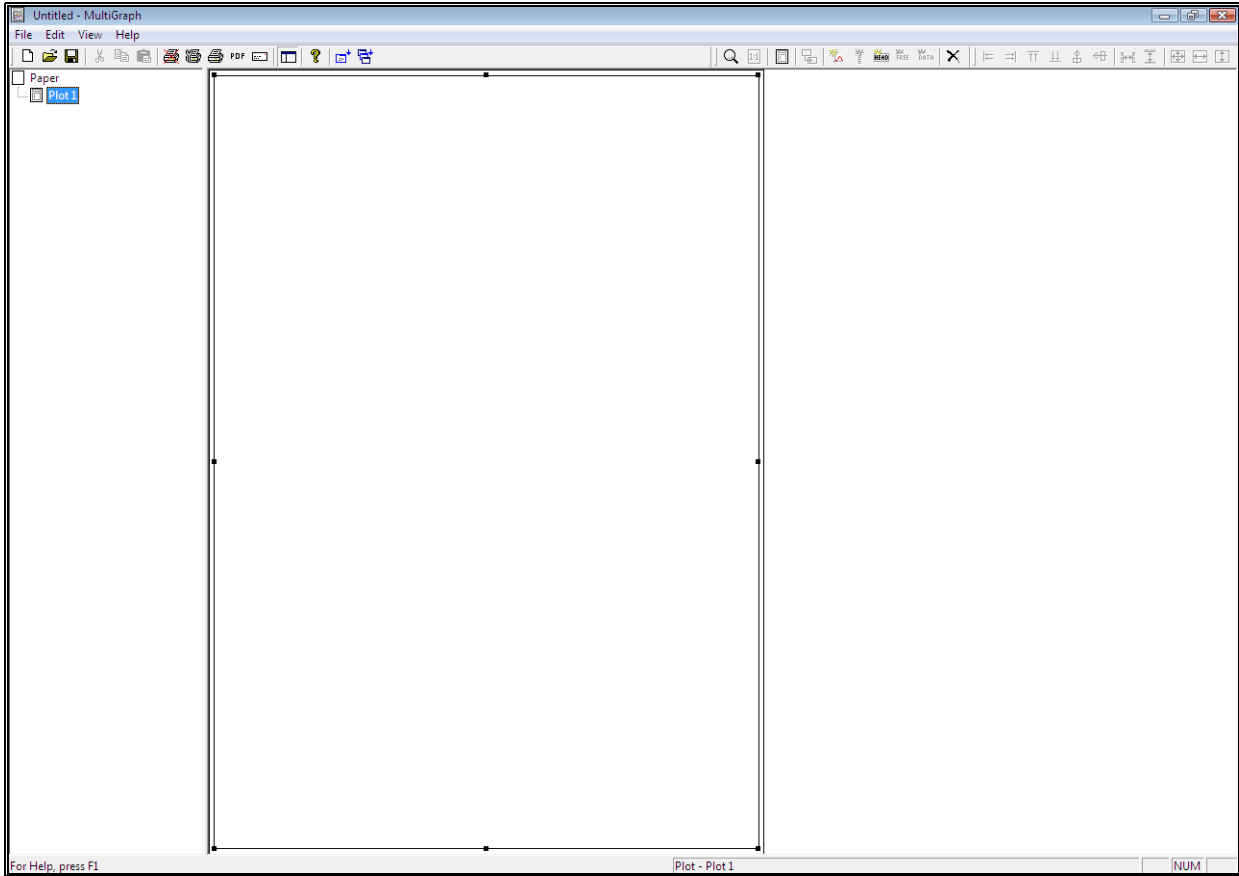
So, if you select a font of **Ariel, size 10** for the plot **all** objects will use this font. You can then change this in the **header** properties to be **Times Roman, size 12** and all objects within that header will take that font. If you then change the plot font to **Lucida, size 10** all objects **except** those in the header will take this font – the header will still use **Times Roman, size 12**.

You can **cancel** the font for the header by picking a font size of 1. All fonts of size 1 are ignored by Multigraph and are taken to mean “use the default font”.

To use the default database table rather than an explicitly selected one, leave the table field blank.

Getting Started

Where do you start? With a blank sheet of paper of course.

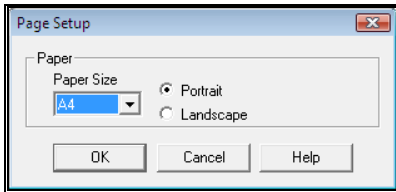


The paper should be set to the size and orientation that you want to use for your plot – basically the size should be the size of the paper it will be printed out to, for on-screen displaying it is irrelevant.

The Plot should then be resized and positioned to fill whichever part of the paper is relevant, probably the whole sheet but margins can be left around it, especially if the plots are usually printed onto header paper or pre-printed forms.

Plot

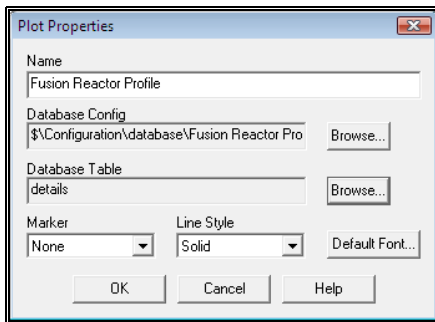
Select the paper size and orientation that you want to use by selecting **Page Setup** from the **File** menu.



For this example we choose to use A4 Portrait.

Next select the properties to be used for this plot. Do this by clicking on the plot in either the left or right panes and then right-clicking the same object.

Properties are propagated down to the objects placed below each level. Whatever properties we choose for the plot will become the defaults that are used for the graphs, axes, headers etc. that are added to the plot.



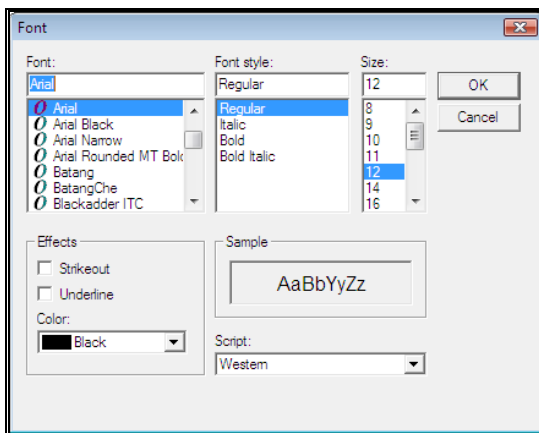
The plot **Name** is not really important, it is only used on the left hand pane as a label.

The **Database Config** is the name of the configuration file that describes the layout of the database that will be used for this plot. Use the **Browse** button to locate the configuration file (you will have created this file with **Access Database Configuration** – see elsewhere in this manual).

The **Database Table** is the name of the table within the database that is going to form the basis of the graphing. It is possible to extract data from other tables within the database, but the name of the table from which most data will be taken should be given here.


The **Marker** and **Line Style** dropdown lists allow you to select the line style and marker to be used with this plot. It's quite likely that you will not select anything here, and just go with a solid line and no marker. Line styles and markers are most useful when you're doing overlay plots, and the overlay plotting facility will allow you to select the line style and marker for each plot when you run it.

The **Default Font** button allows you to choose the font that will be used to propagate into the other items on this plot (explained earlier).



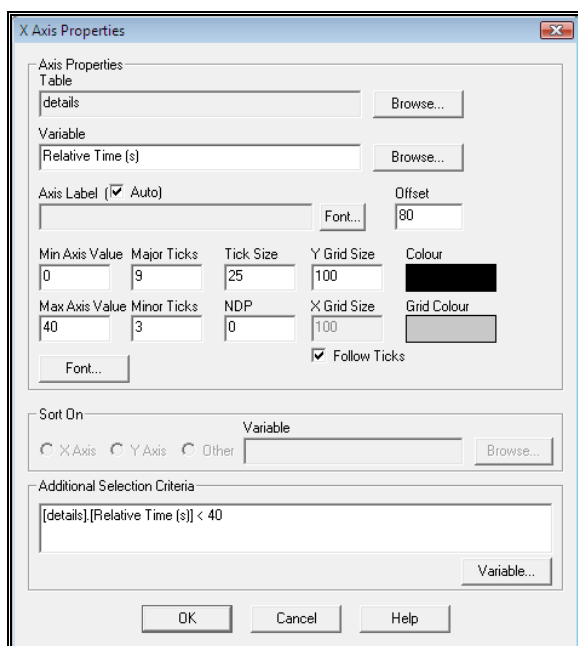
New Graph

The first thing that needs to be done is a new graph added to the plot; this can be done by pressing the new

graph button  located on the menu bar. This should place a graph area onto the plot and if you right click on the graph area the X-axis menu will appear.

X-axis

Graphs come with an X-axis already attached to them. To modify the properties of the X-axis Click on an X-axis object in either the right or left pane and then right click to display the X-axis Properties dialog.



- **Table.**
Select a database table if required. If left blank it will use the default table chosen in the plot properties.
- **Variable**
Select the data element from the selected table that will be used as the X-axis variable.
- **Axis Label.**
This used to label the X-axis and if **Auto** is selected it will use the variable name, otherwise you should enter the label to be used into the Axis Label field.
- **Font.**
Select the font to be used, if required. If left blank (or with size set to 1) it will use the default font chosen in the plot properties.
- **Offset.**
This value is an amount by which the axis label will be offset (downwards) from the actual axis. The value approximates to 10th of mm.
- **Min Axis Value/Max Axis Value.**
These specify the range of the axis. All data values read for the variable specified will be scaled against these values. Any values that fall outside the axis limits will still be scaled and drawn in their appropriate positions, which might be off the graph.
- **Major Ticks/Minor Ticks.**
These values are the number of tick marks that will be drawn on the axis, including the end positions.
- **Tick Size** is the length of the major tick marks, the minor ticks are automatically scaled to be ½ of that length.
- **X/Y Grid Size.**
This is the size of any grid (graticule) that is drawn on the graph.
- **Follow Ticks.**
If this is checked the grid will be at the same axis positions as the major tick marks.
- **Colour.**
This is the colour of the Axis, and its labels.
- **Grid Colour.**
This is the colour of the grid that is displayed on the graph.

- **Additional Selection Criteria.**

This section allows you to specify a rule that will be used when extracting the data. This takes the form of a basic SQL statement, such as [Test Details].[Speed (rpm)] > 500 AND [Test Details].[Speed (rpm)] < 6500. Data will only be extracted for data points that match these selection criteria, even if Speed (rpm) is not the name of the variable being used on the X-axis.

Y-axis



Each X-axis comes with one Y-axis automatically attached. To create more Y axes, click on the X-axis in either the right or left pane (in the left pane it will now be called by the name of whichever variable is attached to the axis) and then click the **New Y-axis** button.

Click on an **Y-axis** object in either the right or left pane and then right click to display the **Y-axis Properties** dialog.

The properties are almost identical to those of the X-axis, the only difference being:

- **Sort On**

This causes the data to be sorted according to the specified criteria.

- **X-Axis.**

This is the default and means data is sorted on ascending order of x-axis value.

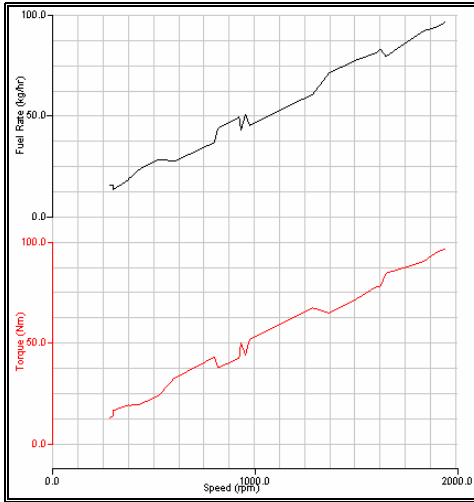
- **Y-Axis.**

Data is sorted in ascending order of the y-axis variable.

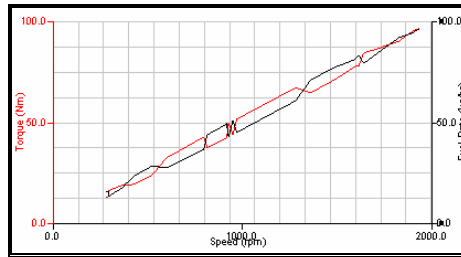
- **Other.**

If this is checked you can select any other variable in the record and have the data sorted by that.

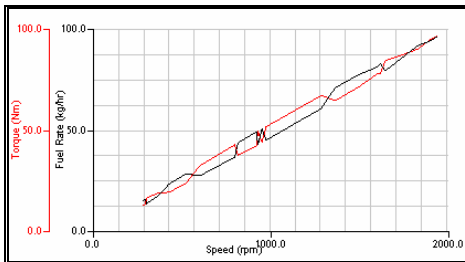
You can move and size the **Y-axis** to be anywhere on the X-axis. Most commonly this would be at one of the sides, and if you have multiple Y axes you can have them stacked one above the other, inside each other, at opposite sides of the X-axis, or any combination of the above. See the examples below.



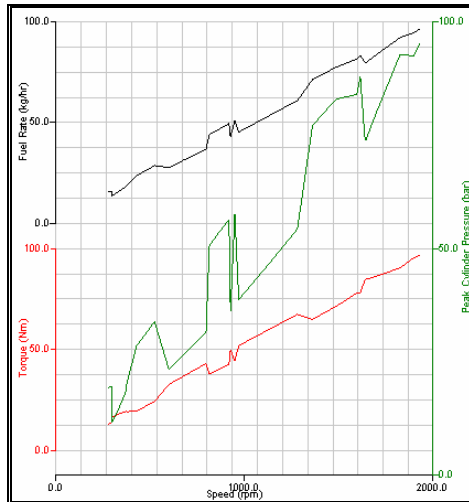
Example 1



Example 2



Example 3



Example 4

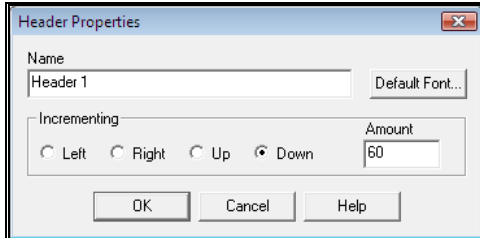
You position the Y axes by dragging them to the left, right, up or down. In addition, when you drag the top or bottom drag handles, the top or bottom of the **Y-axis** will snap to the grid. You can drag one **Y-axis** inside another **Y-axis**, as in the example 3 above.

You move a **Y-axis** to the right hand side of the X-axis simply by dragging it across to the right. When it passes the half way mark it will flip to the opposite side.

Header



You can create a header object for your graph by clicking on the plot object and then clicking on the **New Header** button. The header object will be created in the top 1/3rd of the plot area. Right click on the header object to adjust any header properties.



- **Name.**
This is just a label for the hierarchy tree in the left pane.
- **Default Font.**
If selected this will be the font that is used for all Header Items placed in this header. Cancel the Default Font by selecting a font size of 1.
- **Incrementing.**
This section specifies which way header items inside this header will be incremented if you do a multiple overplot.
 - **Left/Right/Up/Down.**
This is the direction in which successive repeating headers are incremented.
 - **Amount.**
This is the amount by which the next row/column is offset from the previous one.

Headers for multiple overplots work as follows:

M11E TORQUE CURVE					
Marker	Line Style	Test ID	Test Schedule ID	Start Date/Time	End Date/Time
□	—	Ash31	1	07-06-02 14:27:23	07-06-02 15:28:49
◆	—	Ash32	1	14-06-02 14:38:15	14-06-02 14:40:05

Fixed Text header objects appear once and only once in an overplot, but **Data Text** objects appear once for each overplot. In the example above **Marker**, **Test ID**, etc. are fixed text items and **Ash01**, **Ash02** etc. are data text items. The system extracts the data for the first plot and places it in the first row, it then increments by the correct amount (in this case vertically) and draws the data for the second plot, and so on.

The second and subsequent data text items are placed to the left, right, above or below the first data item – the most common position is below. Subsequent items are spaced automatically, depending on the font size that is being used, but you can specify the amount manually in the **Amount** field.

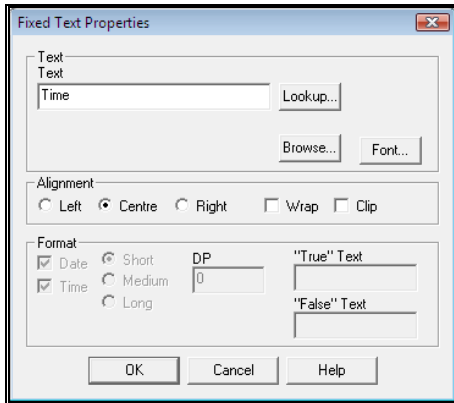
So in summary, for multiple overplots you only place one set of data items on your graph, and when you run the overplots these data items are repeated as many times as there are overplots. The subsequent items are positioned in a direction and by an amount specified in the header properties.

Fixed Text



Fixed Text can also be added to Graph objects.

Fixed text items are added by selecting the header item and then clicking the **New Fixed Text** button. Drag the item to its required position in the header and right click on it to display the properties dialog.

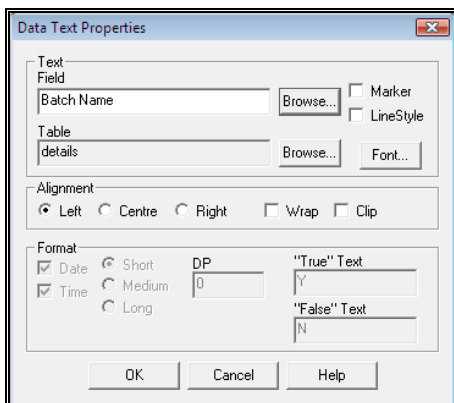


- **Text.**
This is the text that will be displayed, and you can use the **Lookup** button to lookup a data item in the database if you want (it's name in the database will be used as the text).
- **Font.**
This button can be used to choose a specific font for this item, but the font selected for the **header** item will be used as default.
- **Alignment.**
This specifies if the text is to be **left**, **centre** or **right** aligned within its box.
 - **Wrap.**
If this is specified and the text is too wide to fit within the box, it will wrap onto the next line(s).
 - **Clip.**
If this is specified then any text that spills outside of the box will not be drawn, if it is not specified then the full text is always drawn, even if it is too big for its bounding box.

Data Text



Data text items are added by selecting the header item and then clicking the **New Data Text** button. Drag the item to its required position in the header and right click on it to display the properties dialog.



These properties are very similar to **Fixed Text** but with the following additions.

- **Marker.**
If this is selected then the marker style that is being used for that overplot is drawn inside the text box.
- **LineStyle.**
If this is selected then the line style that is being used for that overplot being used is drawn in the text box.
- **Format.**
The various options allowed within the Format group vary according to the data type of the selected field.
 - **Date.**
Available only for Date/Time data types. If selected the date is displayed in the specified format.
 - **Short.**
The date is shown as in 31-12-04.
 - **Medium.**
The date is shown as in 31-12-2004.
 - **Long.**
The date is shown as in 31-Dec-2004.
 - **Time.**
Available only for Date/Time data types. If selected the time is shown (in the form hh:mm:ss).
 - **DP.**
Available only for real data type. This determines the number of decimal places to be shown.
 - **True/False Text.**
Available only for Boolean data types. Select the text to be displayed to show the variable's state.

Other Functions

Zoom



Click the **zoom** button and then click onto the screen. The image will expand to be twice the size that it was. (At present it doesn't centre the zoom on the position that you click at, so you will need to scroll to see the correct portion of the display).

Full Size



Click this button to return to normal sized viewing.

Outline View



Outline view is an alternative way of viewing the plot, designed to make it simpler to see which objects occupy which space on the plot. Clicking this button causes all objects to be drawn as boxes that represent their full size and position on the screen. Click the button again to return to a normal view.

Duplicate



Clicking this button causes an exact duplicate of the current object to be made. If this object is a parent object that contains sub-objects, all of the sub-objects are duplicated too (and any sub-objects that they may contain). The duplicated object becomes the current object, so if you go directly to moving the object, you can know that it is the duplicate that you are moving.

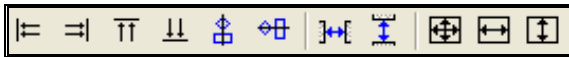
Delete



Clicking this button has the same effect as pressing the delete key, which is to delete the currently selected object. If the object has sub-objects, they are deleted too.









Be careful, there is no Undo facility yet.




Alignment



The alignment facilities are the same as those within Display Builder. Most work when there are two or more objects selected the two *spacing* functions work only when 3 or more objects are selected.

From left to right these are:

- **Align Left.**
 The left hand edges of all objects are aligned with the left hand edge of the last object selected.
- **Align Right.**
 The right hand edges of all objects are aligned with the right hand edge of the last object selected.
- **Align Top.**
 The top edges of all objects are aligned with the top edge of the last object selected.
- **Align Bottom.**
 The bottom edges of all objects are aligned with the bottom edge of the last object selected.
- **Align Centre Horizontal.**
 The centre of all objects are aligned with the centre of the last object selected (objects are moved horizontally).
- **Align Centre Vertical.**
 The centre of all objects are aligned with the centre of the last object selected (objects are moved vertically).
- **Space Horizontally.**
 The objects are spaced horizontally so that there is an equal distance between them (the gap between their edges is equal). The objects at the left and right extremities are not moved.
- **Space Vertically.**
 The objects are spaced vertically so that there is an equal distance between them (the gap between their edges is equal). The objects at the top and bottom extremities are not moved.

- **Same Size.**
 Make all objects the same size as the last selected object.
- **Same Width.**
 Make all objects the same width as the last selected object.
- **Same Height.**
 Make all objects the same height as the last selected object.

Hierarchy



This button toggles the hierarchy display (the left hand pane). When the button is *out* the display is not shown, when it is *in* the display is shown.

Run



This button runs the X-Y plot. It displays a file chooser dialog that enables you to select the database that contains your data (at present the system should be running as 1 batch per database, or *separate databases* as it's known within Prodigy). Multigraph then extracts the data from the correct variables in the correct tables, applies any secondary criteria and/or sorting and draws the plot.

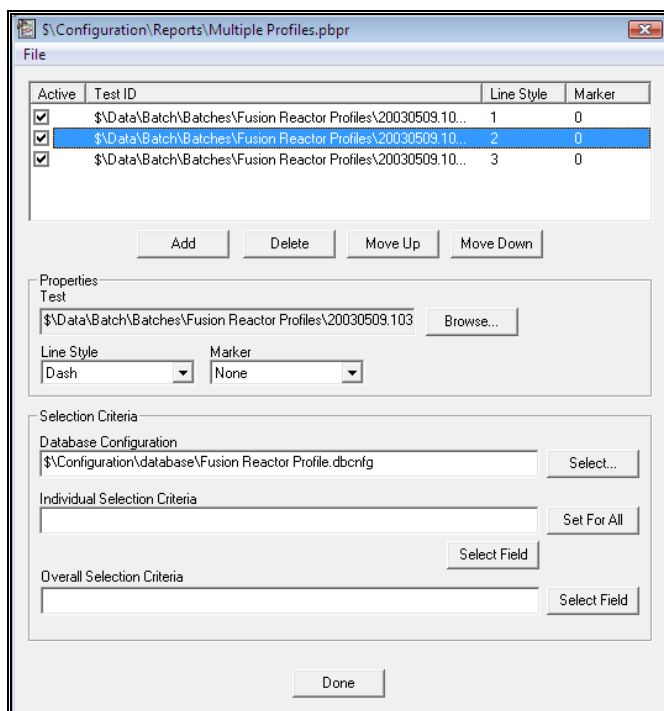
Run Multiple



This button runs a multiple overplot. The overplot definition must be specified first with the separate program **Multigraph Selection** (see next section). When you click this button a file chooser dialog is displayed that allows you to select the Multigraph definition to be run. The actions it then takes are the same as for **Run** except that it collects data and draws graphs for multiple data sets. Any header data is drawn for each separate data set, subsequent sets being offset from the first set by the increment amount specified.

Each graph is drawn in the line style and with the marker type specified in the Multigraph Selection.

Multigraph Selection



The Multigraph selection screen allows you to select a number of X-Y plots that are to be run simultaneously as overplots. It allows you to select a number of different data sets and to specify the line style and marker to be used for each one, as well as some additional selection criteria.

You can open an existing selection or save the current selection by accessing the **File** menu in the top left of the dialog.

Use the **Add** or **Delete** buttons to add or delete extra data sets from the configuration, and use the **Move Up** and **Move Down** buttons to select the order of the data sets. The order in which they are plotted may be important.

Select the database to be used by clicking the **Browse** button and selecting a database.

Choose a **Line Style** and **Marker** type to be used with this data set from the drop down lists.

There are two types of selection criteria – ones that apply to individual databases and one that applies to the data sets as a whole. So obviously the **Individual Selection Criteria** will be one per database, and the **Overall Selection Criteria** will be one per configuration. There is the option with Individual Selection Criteria to apply the currently selected criteria to all other databases in this configuration.

The **Database Configuration** field is required only to allow you use the **Select Field** buttons on the selection criteria fields.

Saved Multigraph Selection configurations can be run from Multigraph to produce overplotted graphs.

Statistical Process Control (SPC)

An Overview Of SPC

SPC is a technique for improving quality within a process by statistically analysing the variables within the process and using this data to identify problem areas. The benefits to be gained from SPC include

- Improved product consistency.
- Improved customer satisfaction.
- Reduction in scrap.
- Reduction in rework.

When selecting variables to be monitored when using SPC, it is usual to begin with those variables that are a measure of the quality of the product (those variables of most concern to the customer). These may be things such as the weight, size, density, resistance, colour, etc. of the finished product. Monitoring these variables will give an indication of overall process performance and will highlight the need for the further application of SPC to the variables involved in the manufacture of the product.

It is crucial to minimise the accumulation of variability within a process, so when applying SPC you should start as early in the process as possible – beginning with the raw materials if possible. A highly variable raw material (e.g. the grade or purity of the material) may be introducing factors into the process that can not be corrected further down the production line (at least economically).

The measurement of some values, particularly those relating to the quality of the finished product, may be beyond the scope of the Prodigy system. Indeed, some measurements may come from laboratory tests. Prodigy SPC allows both data collected by Prodigy and data obtained from external sources to be analysed.

The Prodigy SPC Facility

The Prodigy SPC facility is based on *Projects* and *Runs*.

A **Project** basically specifies the variable that is to be analysed, along with other factors such as which signals determine when the data to be analysed is valid, default charting options and specification limits.

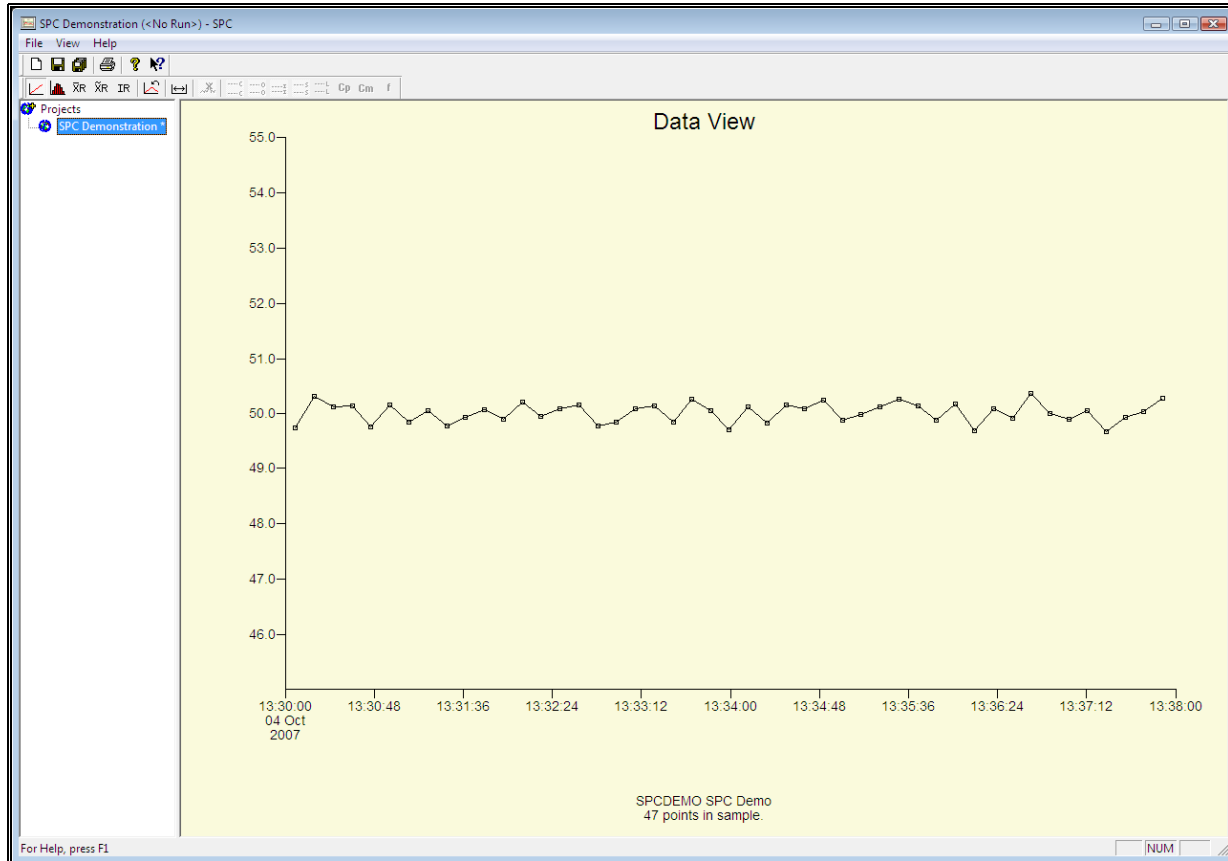
A **Run** is a sample of data for the variable specified in the Project. A Run is taken over a specific period of time (maybe during a batch run, a shift, a day or any period you require). Many Runs may be associated with a given Project to build up a complete record of the Process and allow comparisons to be made.

SPC integrates with Historical Trending, allowing an SPC analysis to be made of any data that is being viewed on a Historical Trend at the press of a button. It also works stand-alone, allowing recorded data for any signal to be analysed and also allowing independently produced data to be analysed via the data entry screen or from the CSV import facility.

The Parts Of The SPC Program

The Main Display

The main display of the SPC facility is shown below.



The display is split into two panes.

- The left-hand pane contains a tree view of all of the SPC *Projects* and their associated *Runs*. The tree can be expanded by clicking on the + signs to the left of an item, or collapsed by clicking on a – sign (note, the + sign changes to a minus and vice-versa, depending on whether the tree is expanded or collapsed). When you click on one of the actual items in the left-hand pane, the right-hand pane displays the data for that Project or Run.
- The right-hand pane displays the data for the currently selected Project or Run. The data may be displayed in one of a number of ways, depending upon the view selected (via the toolbar buttons - see later).

In the picture above, the Run "Batch 1" from Project "Conveyor" has been selected, and the right hand pane is displaying the Data-View for that Run. Notice that at the left hand side of the title bar of the display, the name of the currently selected Project and Run is shown.

The Toolbars

The Standard Toolbar











The Standard Toolbar gives access to the usual range of options associated with document based programs. Reading from left-to-right these are

- **New.**
If there is no Project or Run selected (i.e. the tree item labelled "Projects" is selected), this will create a new Project. If a Project or a Run is currently selected, it will create a new Run. In both cases you will be prompted for the name of the Project or Run. Project names must be unique, Run names must be unique within the currently selected Project but may exist within other Projects.
- **Save.**
Saves the currently selected Project or Run. This button is only enabled if changes have been made to the selected item.
- **Save All.**
Saves all Projects or Runs that have been modified. This button is only enabled if at least one Project or Run has unsaved modifications.
- **Print.**
Prints the contents of the right-hand pane. A standard Windows Print Dialog is displayed that will allow you to select the printer required and any options associated with that printer.
- **Help About.**
Displays the standard Prodigy **About** box, which gives the current package version number and other relevant information.
- **Help.**
Displays a help cursor. While this cursor is displayed, click on a button or pane within the SPC display for help on that item.

The Charts Toolbar



The Charts Toolbar gives access to various display and configuration options within SPC. Reading from left-to-right these are

- **Data View.**
 The current data will be shown in its raw form.
- **Distribution Histogram.**
 The current data will be shown as a distribution histogram.
- **Average & Range.**
 The current data will be shown as an Average and Range chart (sometimes referred to as an Xbar-R chart).
- **Median & Range.**
 The current data will be displayed as a Median and Range chart.
- **Individual & Range.**
 The current data will be displayed as and Individual & Range chart.
- **Refresh Data.**
 The data will be re-read. This option may be used whenever the data that is being displayed may have changed for some reason (e.g., if the data is being taken from a CSV file, the file may have been re-created by the external source).
- **Data Range.**
 Displays a dialog that allows the user to select the timescale over which data will be viewed.
- **Drop out-of-spec points.**
 Allows the SPC system to automatically drop up to two data points that fall outside of the

control limits that are calculated by the system. When the points are dropped, the control limits are re-calculated without the points that have been dropped.

- **Control limits.**



Causes the control limits calculated by the SPC system to be displayed on the relevant charts.

- **Outer warning limits.**



Causes the outer warning limits calculated by the SPC system to be displayed on the relevant charts.

- **Inner warning limits.**



Causes the inner warning limits calculated by the SPC system to be displayed on the relevant charts.

- **Spec limits.**



Causes the Specification Limits specified by the user to be displayed on the relevant charts.

- **Limits for individuals.**



Causes the limits for individuals calculated by the SPC system to be displayed on the relevant charts.

- **Process Capability.**



Causes the Process Capability (Cp) to be displayed on the Average and Range chart.

- **Machine Capability.**



Causes the Machine Capability (Cm) to be displayed on the Average and Range chart.

- **Statistical Factors.**

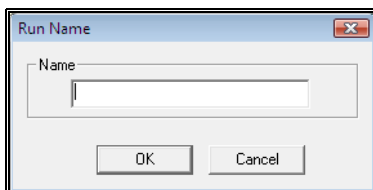


Causes the statistical constants used in the various SPC calculations to be displayed on the relevant charts.

Using the SPC Program

Creating A Project

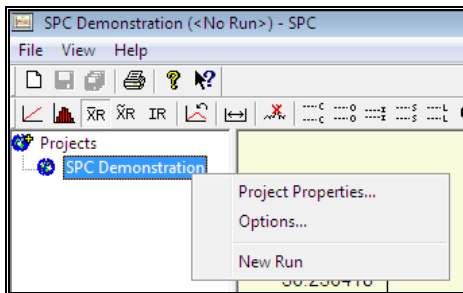
To create a new SPC Project ensure that you have selected the root level of the tree display in the left-hand pane (this is not necessary if there are **no** projects yet) and then press the **New** button on the Standard Toolbar (or select **New** from the **File** menu). A dialog will be displayed requesting a name for the new Project.



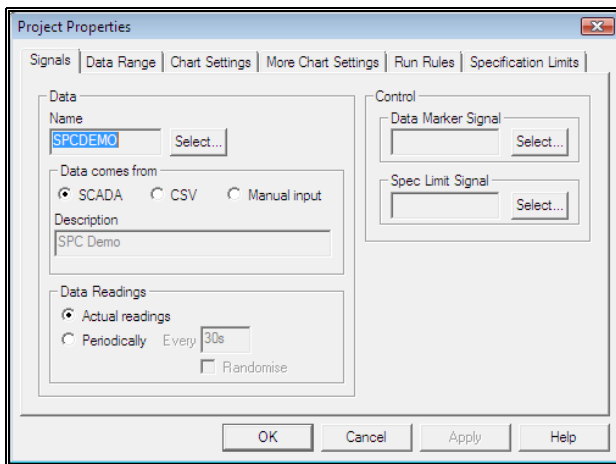
The project name must be unique, if not an error dialog will be displayed. When you press OK the new project will be created. You should then check and modify any properties for this project. The properties selected for a project are used as the default properties for any Runs created within this Project. Most properties may be subsequently modified on an individual basis within each Run.

Project Properties

Right click over the required Project in the left-hand pane and select **Project Properties...** from the popup-menu.



The following properties dialog will be displayed:



This is a multi-tabbed dialog, each tab giving access to a particular range of options. The purpose of each tab is as follows.

Signals

The Signals tab allows you to specify which signal you are going to analyse in this project, together with any signals that define when the signal being analysed is valid. It is also possible to specify that the data will be taken from a CSV file or from Manual Input.

- **Data.**
This section allows you to specify where the data will be taken from.
- **Name.**
Omit this step if the data is to be taken from a CSV file or Manual Input.
Select the name of the signal that is to be analysed by pressing the **Select...** button. This will display the standard SigScan dialog that will enable you to choose the required signal. Note that SigScan will only allow you to select from signals that have a recording strategy associated with them. This is because SPC takes its data from the recorded values for the signal.
- **Data comes from.**
This section describes where the data will be taken from. If **SCADA** is selected, the data will be taken from recordings made by Prodigy.
If **CSV** is selected, the data will be taken from a CSV file. A dialog allowing you to specify the name of the CSV file is provided within **Run Properties...** (see later). This option is for use when data for SPC will be exported from some other package (a spreadsheet or database, for instance).
If **Manual Input** is selected, the data will be entered directly into SPC via a screen that is available within **Run Properties...** (see later). This option is for when data is being generated by some means that the Prodigy system is not monitoring (laboratory tests, for instance).
*The option to enter a **Description** becomes available when data is taken from CSV or Manual*

input. When the data is taken from Prodigy the charts are labelled with the description of the signal, as found in the Signal Database. When the data come from CSV or Manual Input, you must provide the signal description.

- **Data Readings.**

Data readings can be taken in one of two ways:

- **Actual readings.**

Use this if the signal has been recorded with a *regular* recording strategy, that means that recordings are made every *n* seconds or minutes, etc. Do **not** use this option if data is recorded only when it's value changes by a certain amount – that will give inaccurate results.

- **Periodically.**

Use this when the signal has been recorded on change of value. You specify how many seconds gap to use (*n*) and the system finds out the value of the signal every *n* seconds from the start time. Thus, if *n* is 10 seconds and the signal was recorded for an hour but only changed value twice, we will get 360 readings. With **Actual readings** we would have only had 3 readings. This method can be used with data that had regular recording, the frequency given here will be used as an override to give more or less data points than the **Actual readings** method would give.

The **Randomise** button may be checked to force the readings to be taken at a *not quite regular* interval. The actual interval will be the interval specified + or – 20%, i.e. if the interval specified was 10 seconds, the actual interval would be anywhere in the range 8 to 12 seconds. The interval is recalculated for each reading, so the time between successive readings constantly varies.

- **Control.**

This section allows you to specify up to two signals that may be used to control whether or not individual data points within a specified time range may be used by SPC. When SPC reads the data for a given run, it will only consider any points that occur when both of the signals given here are set (i.e. have a value of 1 or greater).

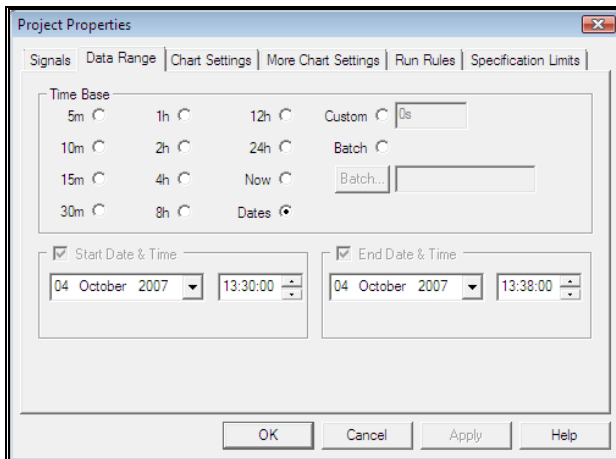
If only one signal is given, then only that one needs to be set. If no signals are given here then all points in the specified range are included in the SPC analysis.

*These two signals **must** be recorded, but in this case it is fine to have them recorded **on change of value**.*

- **Data Marker Signal** is the name of a signal within Prodigy that you are using to specify when the signal that is being analysed is valid. This signal will be a Digital (or Finite State) and the data will be assumed to be valid when the signal value is 1. This signal will be set by some means when the process is running and producing valid data. It will be cleared when the process is not running or not producing valid data. For instance, the signal may be set when a batch is running and cleared when the batch ends. Indeed, you may use the signal here that you are using to indicate the batch status (see the XForms section). Alternatively, the signal may be derived (or set by a SLANG program) such that it is set when the process has been running for a given length of time, or after a certain amount of product has been produced. Any readings take before this period will be ignored as the process was not deemed to be "up to speed".

- **Spec Limit Signal** is the name of a Digital or Finite State signal that is being used to indicate when the process is in a state suitable for the readings to be used for SPC. Its function overlaps somewhat with that of the Data Marker Signal (above). Whereas it is envisaged that the Data Marker signal will indicate whether or not the process is running, the Spec Limit Signal will be cleared when something happens within the process that means you would want to ignore any data being produced for SPC purposes. Such things may include signals going into an alarm state, signal values falling above or below certain levels, and so on (and by that we generally mean signals other than the one being monitored by this SPC Project).

Data Range



This allows you to specify the time scale for which data will be analysed.

- **Time Base.**

Select a time period over which data is to be taken. In most cases the time period is counted forwards from the start date or backwards from the end date, depending on whether a **Start Date & Time** or an **End Date & Time** is selected.

There are standard settings ranging from 5m (5 minutes) to 24h (24 hours). In addition there are some special options.
- **Now.**

When Now is selected a **Start Date & Time** must also be given. Data is read from that time up to the current time. This option is most commonly used when an analysis is being performed on a process that is still running. More and more data values will be being generated while the SPC program is in use. The **Refresh Data** button on the **Charts Toolbar** may be used to cause SPC to re-read the data to include any new values that have been generated.
- **Dates.**

When this option is selected both a **Start Date & Time** and an **End Date & Time** must be given.
- **Custom.**

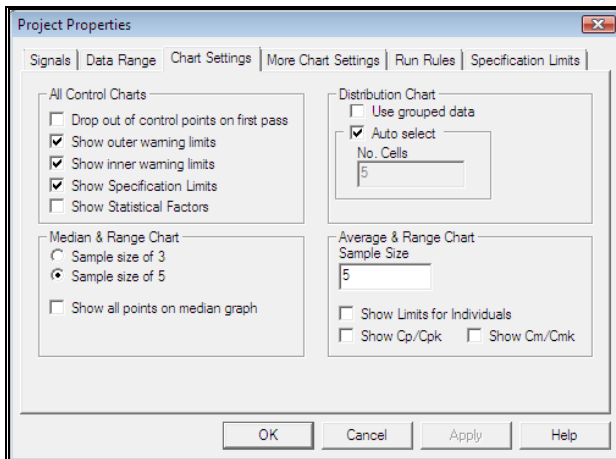
Enter any time period you require as a standard Prodigy time period string (e.g. 1h24m16s = 1 hour, 24 minutes and 16 seconds).
- **Batch.**

Use the **Batch...** button to select a batch from the Batch database. The **Start Date & Time** and the **End Date & Time** will be set to be the start and end times of the batch that is selected.
- **Start Date & Time.**

If this box is checked then you should enter a Start Date & Time. The data range will be counted forwards from this time for the range specified in the section above.
- **End Date & Time.**

If this box is checked then you should enter an End Date & Time. The data range will be counted backwards from this time for the range specified in the section above.

Chart Settings



This allows you to specify various chart settings that will act as defaults for Runs created within this project.

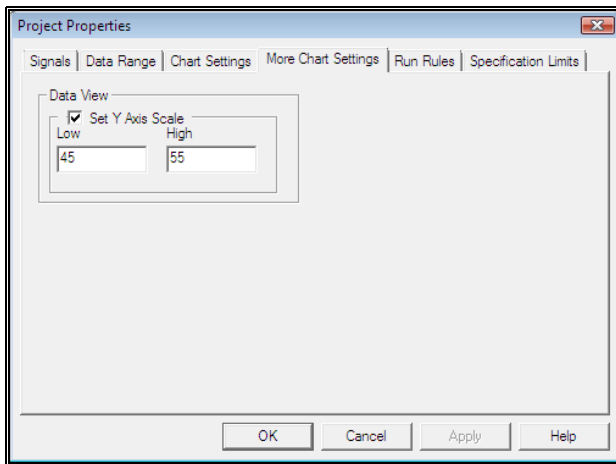
- **All Control Charts.**
The options in this section apply to all relevant charts.
- **Drop out-of-control points on first pass.**
If this option is checked the following actions will occur.
When SPC has calculated the Range Chart control limits for an Average & Range chart it will look for any ranges that fall outside of these control limits. If one or two points fall outside the control limits, these points are eliminated from the data and the control limits recalculated. If there were more than two points, or if after recalculation there are further points outside of the control limits, the data will be flagged as having *out-of-control points* and will be deemed to be *not in statistical control*.
When SPC calculates the Average Chart control limits it performs the same checks on points falling outside these limits as it did for the Range Chart and will eliminate up to two out-of-control points.
In both cases, if any out-of-control points are eliminated, this is indicated on the charts. You should look to the process to see if you can attribute a reason for these points and take any action available to prevent them recurring.
- **Show outer warning limits.**
The outer warning limits are used during run-rule checking. Checking this option causes the outer warning limits to be drawn on the charts.
- **Show inner warning limits.**
The inner warning limits are used during run-rule checking. Checking this option causes the inner warning limits to be drawn on the charts.
- **Show Specification Limits.**
The Specification Limits are the tolerances to which the process should adhere. These are used in the determination of machine and process capability and checking this option causes them to be drawn on the charts.
- **Show Statistical Factors.**
Checking this option causes the statistical factors used in the calculation of the control limits and machine and process capability to be printed on the charts. These factors vary with the sample sizes selected elsewhere on this dialog.
- **Median & Range Chart.**
This allows you to set options for the Median & Range Chart.
The Median & Range chart is a control chart where groups of successive points are taken and the median value of each group is taken, along with the difference between the largest value in the group and the smallest. These values are then plotted on two charts sharing a common x-axis. The median values are plotted on the upper part of the chart and the range values are plotted on the lower part. Each chart has a set of control lines calculated for it based on the data that has been used. The points are compared against these control lines to see if there are any run-rule violations or out-of-spec data points. Any anomalies that are found are reported on the chart and the anomalous points are drawn in a different colour.

- **Sample Size of 3.**
Sets the sample size to 3. Data readings are taken as groups of 3 successive points and the median and range are calculated for each group.
- **Sample Size of 5.**
Sets the sample size to 5. Data readings are taken as groups of 5 successive points and the median and range are calculated for each group.
- **Show all points on median graph.**
If this option is checked all of the points in each sample are plotted when the median chart is drawn. If it is not checked, only the median point is plotted.
- **Distribution Chart.**
This allows various options for the Distribution Histogram to be selected. The Distribution Histogram is as follows. The overall range of the data is calculated (i.e. the difference between the largest and the smallest values in the sample). This range is then divided into a number of "cells". Each point in the sample is then checked to see which cell it would fall into and the total number of points that would fall into each cell is determined. A histogram is then drawn from the resulting data. In order for the data to be suitable for an SPC analysis it should follow a **normal** distribution. If not, adjustments must be made to the process to bring the data into line with a normal distribution. If this is not possible then any conclusions drawn from the SPC analysis on this data will be invalid.
- **Use grouped data.**
If this option is checked the histogram is produced using grouped data rather than the individual data points. The data is taken as being groups of successive points, the number of points in each group being determined by the **Sample Size** for the **Average & Range Chart**. The average value of each group is then used to produce the distribution histogram. Use this option if the raw data does not appear to follow a normal distribution. If the grouped data does not follow a normal distribution either, the data is not suitable for an SPC analysis. Examine the process to determine if there is any action that you can take to remedy the situation.
This option is offered because the Average & Range charts, and hence the C_m and C_p values produced, are based on grouped data. Your first recourse should be to the shape of the ungrouped distribution, though.
- **Auto select.**
If this option is checked the number of cells that the distribution histogram is divided into is calculated automatically. This number varies with the number of points in the sample (adjusted to account for whether or not the **Use grouped data** option has been checked). The number of points in the sample gives rise to the following number of cells.
< 50 points, 6 cells.
< 100 points, 8 cells.
< 150 points, 10 cells.
>= 150 points, 12 cells.
If the Auto select option is not checked, the histogram is divided into whatever number of cells you specify in the box provided, within the range of 5 to 50.
- **Average & Range Chart.**
This allows you to set options for the Average & Range Chart. The Average & Range chart is a control chart where groups of successive points are taken and the average value of each group is taken, along with the difference between the largest value in the group and the smallest. These values are then plotted on two charts sharing a common x-axis. The average values are plotted on the upper part of the chart and the range values are plotted on the lower part. Each chart has a set of control lines calculated for it based on the data that has been used. The points are compared against these control lines to see if there are any run-rule violations or out-of-spec data points. Any anomalies that are found are reported on the chart and the anomalous points are drawn in a different colour. You will notice that this is very similar to the Median & Range chart. The Average & Range chart goes a stage further because the values calculated during its production can be used to generate machine and process capability indices (**C_m** and **C_p**).
- **Sample Size.**
Enter the number of successive points that will be used to form each group. The number must be in the range 2 to 6.
- **Show Limits for Individuals.**
Limits for individuals are two control values that are used to determine the machine capability.

If these values fall within the specification limits, the machine or operation may be judged to be capable. Checking this option causes the Limits for individuals control lines to be drawn on the Average & Range chart.

- **Show Cp/Cpk.**
The **Process Capability** indices **Cp** and **Cpk** will be shown on the chart if this option is checked.
- **Show Cm/Cmk.**
The **Machine Capability** indices **Cm** and **Cmk** will be shown on the chart if this option is checked.

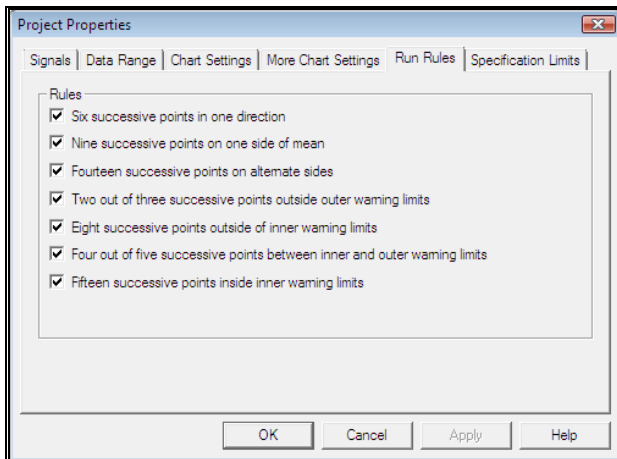
More Chart Settings



This is a continuation of the features offered by the previous tab (there are too many options to fit comfortably onto one tab).

- **Data View.**
 - **Set Y-axis Scale.**
If this option is checked you will be able to enter the low and high values to be used on the Y-axis scale of the Data View. If you do not check this option, the graph will be drawn against a scale taken from the signal range in the signal database. Where the data is being taken from a CSV file or from Manual Input, this option must be checked and appropriate values entered, otherwise the chart will not be drawn.

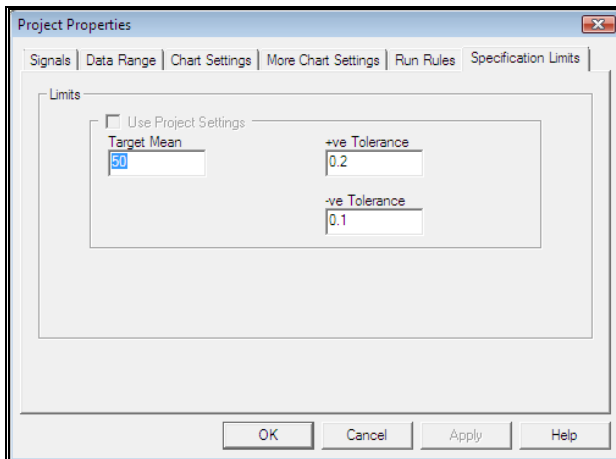
Run Rules



This dialog allows you to select that run-rules are to be applied when determining whether or not the process is in **statistical control**. Normally, all run-rules would be applied, but you may have a requirement to eliminate certain of the rules. The run rules are based on statistical probability. For a normal distribution, the chances of any of these run rules occurring are very, very small. If one does occur within your process, it highlights a problem. Most of the run-rules are self explanatory, though the reasons behind them may not be obvious. As mentioned earlier, it is down to statistical probability.

- **Six successive points in one direction.**
This run-rule checks to see if six points in a row rise or fall. If they do, the process is declared to be out of statistical control.
- **Nine successive points on one side of mean.**
If this pattern occurs, the process is declared to be out of statistical control.
- **Fourteen successive points on alternate sides (of mean).**
If this pattern occurs, it shows that the process is not very steady but is constantly cycling from one side of the mean to the other, indicating that the process is out of statistical control.
- **Two out of three successive points outside outer warning limits.**
If this pattern occurs, the process is declared to be out of statistical control.
- **Eight successive points outside of inner warning limits.**
If this pattern occurs, the process is declared to be out of statistical control.
- **Four out of five successive points fall between the inner and outer warning limits.**
If this pattern occurs, the process is declared to be out of statistical control.
- **Fifteen successive points inside inner warning limits (on either side of mean).**
If this pattern occurs, the process is declared to be out of statistical control.

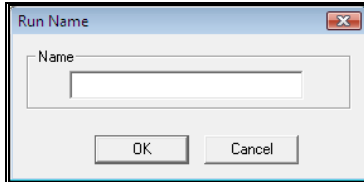
Specification Limits



This dialog allows you to specify the Specification Limits to which the process value being analysed is expected to conform. The Specification limits consist of a **Target Mean** and a **Positive** and **Negative Tolerance**. Often the tolerances are the same for both positive and negative, but sometimes they may differ. A peg intended to fit into a hole may be allowed to be substantially smaller than the mean value, but only fractionally larger. This is because no matter how much smaller it is, it will always fit into the hole. Of course, this tolerance will depend on whether or not it is meant to be a tight fit!

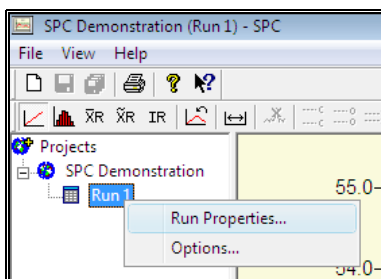
Creating A Run

To create a new SPC Run ensure that you have selected the Project to which the Run is to belong or another Run within that Project. Press the **New** button on the Standard Toolbar (or select **New** from the **File** menu). A menu will be displayed requesting the name of the new Run.



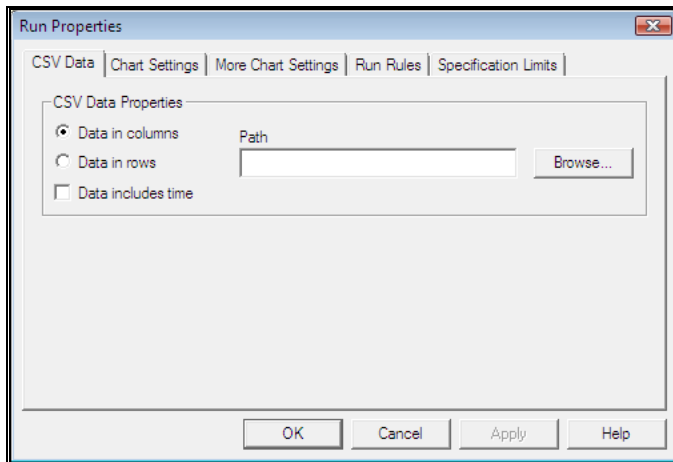
The Run name must be unique within this Project, but there can be runs of the same name within other Projects. When you press OK the Run will be created within the current Project. Most of the properties for the run are copied from those you have set up for the Project. You should modify any properties as required and select a Data Range for this run.

The **Run Properties** dialog can be accessed by right clicking on the Run name in the left-hand pane and selecting **Run Properties...** from the popup menu.



The options within this dialog are exactly as for the Project Properties with the following additional tabs being available:

CSV Data



This dialog allows you to enter the path of the CSV file from which data is to be read. Use the **Browse...** button to select the required file. Select the relevant options as to whether the data is in columns or rows and whether or not it contains a timestamp for each data point. If timestamps are included they should come *after* the data values, either one per row if the data is in columns or as a complete row if the data is in rows. Examples are shown below. Note that the date/time format is fairly flexible, but be aware of any locale settings that could cause a problem (American date formats are different from UK formats).

Data In Rows With Time

750.0,750.1,750.3,749.7,749.97

1-8-99 10:11:12,1-8-99 10:11:13,1-8-99 10:11:14,1-8-99 10:11:15,1-8-99 10:11:16

Data In Columns With Time

750,23-8-99 11:07

750,23-8-99 11:08

750.1,23-8-99 11:09

750,23-8-99 11:10

749.9,23-8-99 11:11

749.9,23-8-99 11:12

750.1,23-8-99 11:13

750.05,23-8-99 11:14

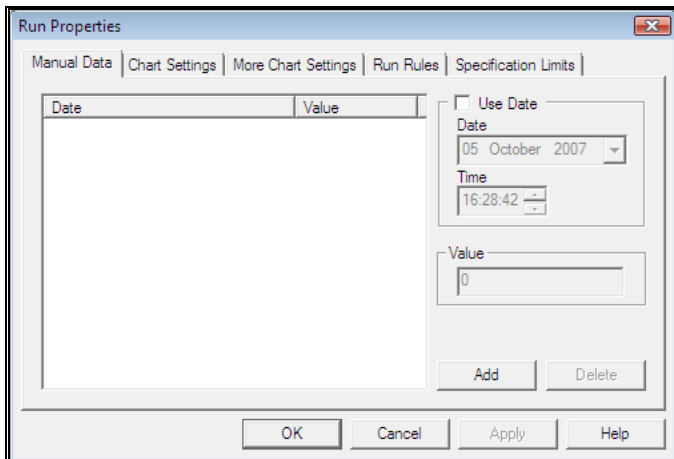
750,23-8-99 11:15

749.9,23-8-99 11:16

749.7,23-8-99 11:17

749.95,23-8-99 11:18

Manual Data



This dialog allows you to enter data directly into Prodigy SPC. The data can be either date-related or merely sequential. If it is date related, the order in which it is entered is not important and it is sorted into ascending date order on exit from the screen. If no timestamp is entered, data is used in the order it appears in the list within the dialog.

To enter new data into the list, click on an existing item (if there is one) and press the **Add** button. A new item will be created *below* the currently selected item, with all details being copied from the current item. (An exception to this is if you select the first item in the list. In this case you will be offered the opportunity to add the new item *above* or *below* the current item).

To delete items from the list, click on the item that you want to delete and then press the **Delete** button.

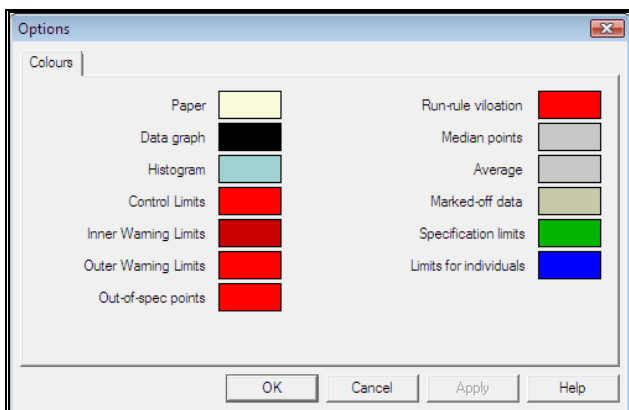
To modify list entries, click on the relevant item in the list and enter a new **Date**, **Time** or **Value**. The entry in the list changes automatically, in line with any changes you make to the fields.

Check or uncheck the **Use Date** box, depending on whether or not you want your data to be timestamped. If you do not timestamp it, the data is regarded as being sequential in the order it appears in the list. The Data View graph will not be labelled with any time axis.

When you exit from the **Run Properties** dialog by pressing **OK** you will be asked if you want to save the Manual Data (if you have made changes to it). If you reply **Yes** the data will be rewritten. If you reply **No** the other changes you have made via the **Run Properties** dialog will be saved, but the Manual Data will *not* be rewritten. If you reply **Cancel** then the **Run Properties** dialog will not be exited and you will have the option to make further changes.

Setting Up Colours

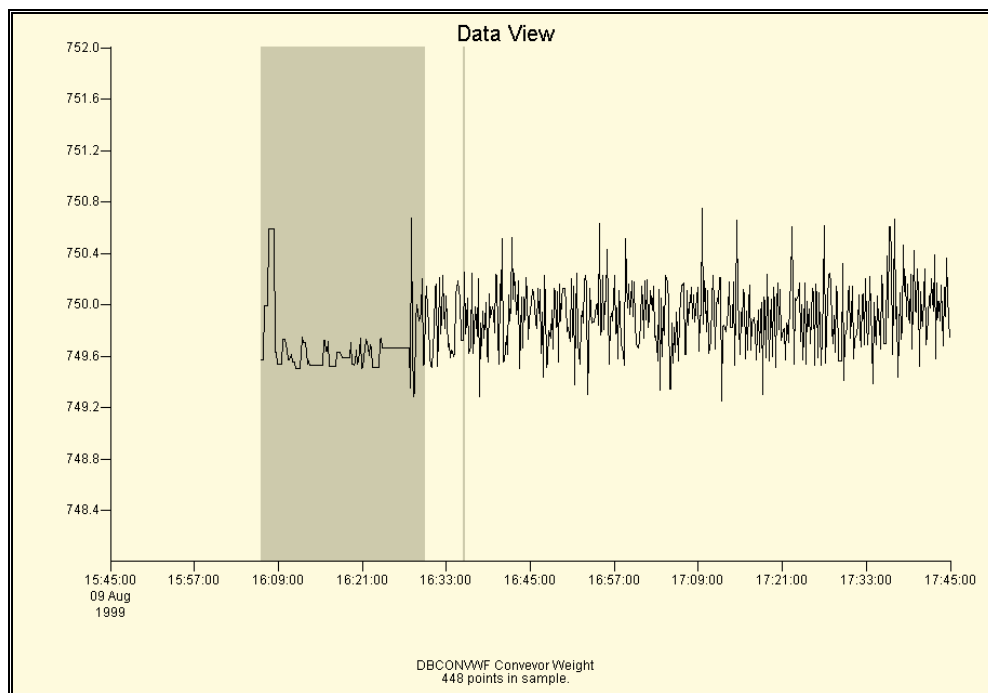
The **Options** dialog can be accessed by right clicking on the Run name in the left-hand pane and selecting **Options...** from the popup menu. This displays the dialog below which allows you to select the colours used in the SPC charts.



Viewing The Data In Prodigy SPC

Prodigy SPC provides a variety of ways to view the data, ranging from raw data views, through distribution charts to fully annotated control charts. The different views are accessed via the **Charts Toolbar** and are explained in detail below.

The Data View

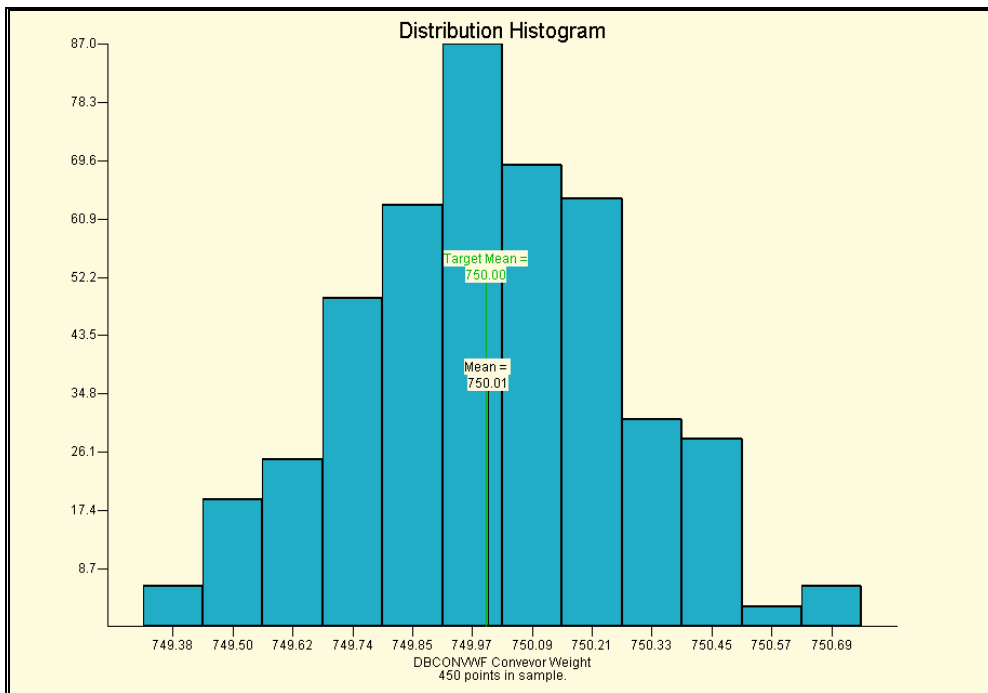


When the Data View is selected, the right-hand pane displays a line graph of the individual values that have been read in. The x-axis is time-ordered and shows the duration for which this Run has been specified (*unless the data has been read from a CSV file or Manual Input, in which case it may optionally be not associated with a particular time. It will still be time ordered, however*).

If **Data Marker** and/or **Spec Limit** signals were specified for the Project, any periods of time which are being excluded will be displayed as a shaded region on the chart. The legend at the bottom of the graph which indicates how many points are in the sample will not include any points that are in the excluded region, nor will such points be included in any other charts or calculations within this SPC run.

In the Data View shown above, we can see that prior to about 16:09 there was no data at all (Prodigy was not running, or the plant I/O was not connected, etc.). After this time there was a period where data was available, but it is flagged as being *invalid* for SPC purposes. This is because the **Data Marker** flag was not set – the process was just warming up and was not in actual production. After this we are getting valid data, except for a short time after 16:33. At this point the **Spec Limit** signal was set to indicate that something was wrong somewhere in the process at this time and that the data taken during that period should not be used for SPC purposes.

The Distribution Histogram



The Distribution Histogram is used to ensure that the data follows a normal distribution. The chart is produced as follows.

The overall range of the data is calculated (i.e. the difference between the largest and the smallest values in the sample). This range is then divided into a number of *cells*, each cell having an upper and lower limit. Each point in the sample is then checked to see which cell it would fall into and the total number of points that would fall into each cell is determined. A histogram is then drawn from the resulting data.

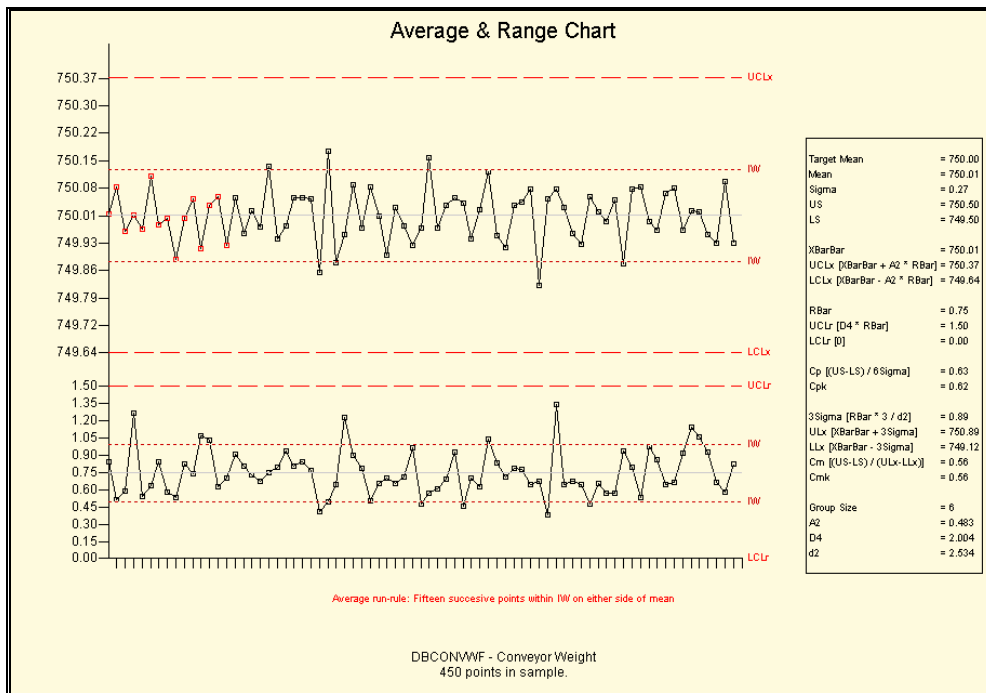
In order for the data to be suitable for an SPC analysis it should follow a *normal* distribution. If it does not, adjustments must be made to the process to bring the data into line with a normal distribution. If this is not possible then any conclusions drawn from the SPC analysis on this data will be invalid.

The Distribution Histogram can be produced from either the raw data or from *grouped* data. The grouped data histogram can be used as a double check that the data is normally distributed. The raw data may not quite follow a normal distribution, but if the grouped data histogram does, then it is usually OK to proceed with the SPC analysis.

To produce a histogram from grouped data the average value of the values in each group is used to produce the histogram. Each group is made up from a number of successive points, this number being equal to the sample size for the **Average & Range Chart**. These options are set via the **Run Properties**.

The Distribution Histogram may be optionally overlaid with the inner and outer warning limits, the control limits and the spec limits (the inner and outer warning limits and the control limits are the mean $\pm 1, 2$ and 3 standard deviations). The Mean and the Target Mean are always shown.

The Average & Range Chart



The Average & Range chart is the main workhorse of the SPC analysis. It is used to check that the process is in statistical control and to provide machine and process capability indices. The Average & Range chart consists of two charts in one, sharing a common x-axis, these charts being the Average Chart and the Range Chart.

The Average & Range chart is produced as follows.

- The data is divided into groups formed from successive points, the number of points in each group having been determined in the **Run Properties**.
- The **Average** of each group is calculated, along with the **Range** (the difference between the largest value in the group and the smallest).
- The average of all of the group averages is calculated. This acts as the centre line for the Average Chart.
- The average of all of the ranges is calculated. This acts as the centre line for the Range Chart.
- Control limits are calculated for each of the charts.
 - The **Upper Control Limit** for the Range Chart is known as **UCLr** and is calculated as being $D_4 * (\text{the average range})$.
 D_4 is a statistical constant that varies with the group size.
 - The **Lower Control Limit** for the Range Chart is known as **LCLr** and is assumed to be 0.
 - The Range Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Range Chart control limits are recalculated. (The out-of-spec points will **not** be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).
 - The **Upper Control Limit** for the Average Chart is known as **UCLx** and is calculated as being (the average of averages) + $A_2 * (\text{the average range})$.
 A_2 is a statistical constant that varies with the group size.
 - The **Lower Control Limit** for the Average Chart is known as **LCLx** and is calculated as being (the average of averages) - $A_2 * (\text{the average range})$.
 A_2 is a statistical constant that varies with the group size.
 - The Average Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Average Chart control limits are recalculated. (The out-of-spec points will **not** be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).

- Both charts are checked for out-of-spec points and run-rule violations, starting with the Range Chart. If any violations are detected they are indicated on the chart. The process is not in statistical control if any violations occur.
- The **Upper Limit for Individuals** is known as **ULx** and is shown only on the Average Chart. It is calculated as being
 $(\text{the average of averages}) + 3 * d_2 / (\text{the average range})$.
d₂ is a statistical constant that varies with the group size.
- The **Lower Limit for Individuals** is known as **LLx** and is shown only on the Average Chart. It is calculated as being
 $(\text{the average of averages}) - 3 * d_2 / (\text{the average range})$.
d₂ is a statistical constant that varies with the group size.
- The **Machine Capability Index** is known as **Cm** and is calculated as being
 $((\text{Upper spec limit}) - (\text{Lower spec limit})) / (\text{ULx} - \text{LLx})$.
 In addition, the **Cmk** value is calculated. The Cm value takes account of the *process mean* being off centre from the *target mean*. The Cm value can show that the machine is *capable* while the Cmk value shows that it is *not capable*. In this instance, the indices are telling us that the machine is *capable of being capable* but because the process mean is off centre, it is not currently actually capable.
- The **Process Capability Index** is known as **Cp** and is calculated as being
 $((\text{Upper spec limit}) - (\text{Lower spec limit})) / (6 * \text{standard deviation})$.
 In addition, the **Cpk** value is calculated. The Cp value takes account of the *process mean* being off centre from the *target mean*. The Cp value can show that the process is *capable* while the Cpk value shows that it is *not capable*. In this instance, the indices are telling us that the process is *capable of being capable* but because the process mean is off centre, it is not currently actually capable.

Machine Capability v Process Capability

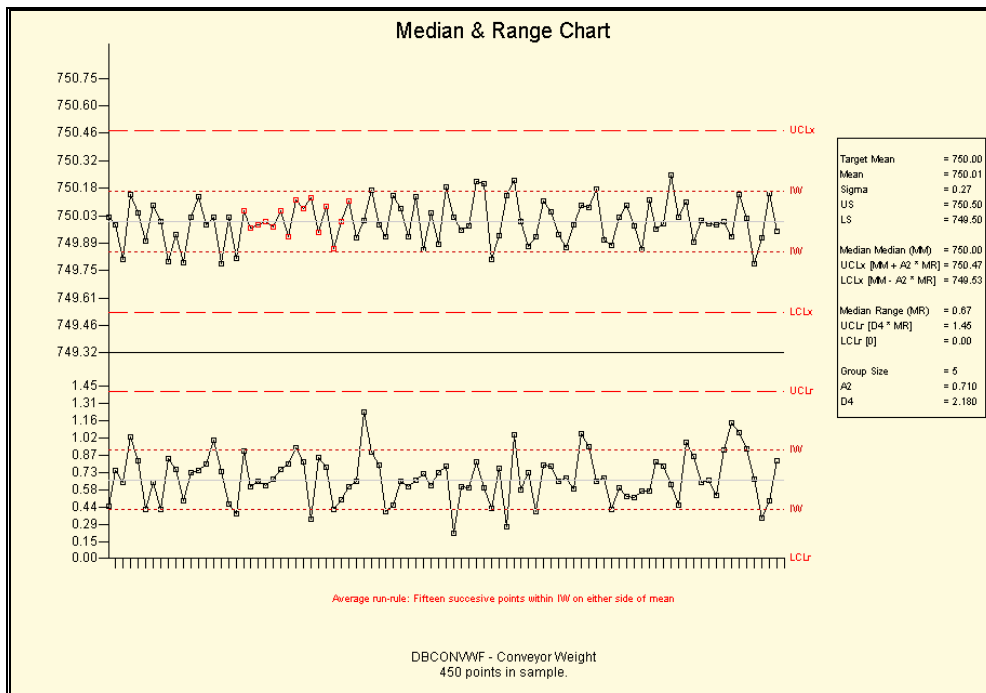
Machine capability and process capability are very similar but are based on slightly different premises. Both can be affected by the quantity of data used in their calculation, and the timing of such data.

When calculating *Machine Capability* the data being used in the calculation should be as free from external factors as is possible, i.e. any variations in the data should come from the machine itself. In order to achieve this it is usual for the data to be taken over a relatively short period of time and to be taken at a time when other factors are known to be as constant as possible. For example, only use data where the raw material is known to come from one particular batch, or where only one operator is working a machine. Do not use data where it is known that a fresh batch of raw material was added part way through the run or where the operators changed shifts, otherwise variation in the process value may possibly be attributed to changes in raw material quality, operator skill etc.

When calculating *Process Capability* it is necessary to take data from a longer period of time -30 days is often recommended. This ensures that other external factors *are* taken into account. The capability of the process *will* depend on the quality of raw material, the skill of a machinist, the accuracy of the temperature controller, etc.

The Process Capability is calculated using the standard deviation of the raw data whereas the Machine Capability is calculated using the average range, which reflects mainly machine variation.

The Median & Range Chart



The Median & Range chart is an alternative way of viewing the data. It can be used when the operation being monitored is

- Known to be distributed normally.
- Is not very often disturbed by assignable causes.
- Is easily adjusted by the operations staff.

The Median & Range chart is produced as follows.

- The data is divided into groups formed from successive points, the number of points in each group having been determined in the **Run Properties**.
- The **Median** of each group is calculated, along with the **Range** (the difference between the largest value in the group and the smallest).
- The median of all of the group medians is calculated. This acts as the centre line for the Median Chart.
- The median of all of the ranges is calculated. This acts as the centre line for the Range Chart.
- Control limits are calculated for each of the charts.
 - The **Upper Control Limit** for the Range Chart is known as **UCLr** and is calculated as being $D_{\sim 4} * (\text{the median range})$.
 $D_{\sim 4}$ is a statistical constant that varies with the group size.
 - The **Lower Control Limit** for the Range Chart is known as **LCLr** and is assumed to be 0.
 - The Range Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Range Chart control limits are recalculated. (The out-of-spec points will **not** be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).
 - The **Upper Control Limit** for the Median Chart is known as **UCLx** and is calculated as being (the median of medians) + $A_{\sim 2} * (\text{the median range})$.
 $A_{\sim 2}$ is a statistical constant that varies with the group size.
 - The **Lower Control Limit** for the Average Chart is known as **LCLx** and is calculated as being (the median of medians) - $A_{\sim 2} * (\text{the median range})$.
 $A_{\sim 2}$ is a statistical constant that varies with the group size.
 - The Median Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Median Chart control limits are recalculated. (The out-of-spec

points will *not* be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).

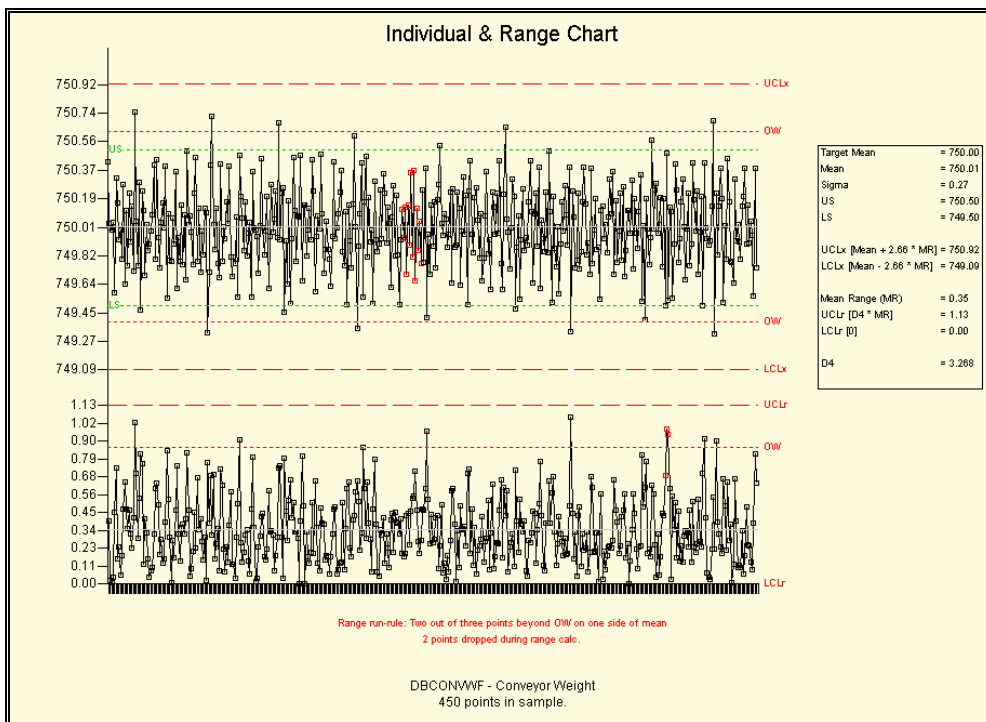
- Both charts are checked for out-of-spec points and run-rule violations, starting with the Range Chart. If any violations are detected they are indicated on the chart. The process is not in statistical control if any violations occur.

The medians plotted on the Median chart are actual values (in fact, the chart can be configured to show **all** of the data values) and therefore these values can be compared against the specification and control limits. Remember though, that corrective action is not required if individual points fall outside of the control limits, only if median or range values fall outside of these limits or if run-rules are violated. In both of these cases, Prodigy SPC will indicate that there is a problem.

The Cm and Cp values cannot be calculated from data provided by the Median & Range chart.

Within the automated environment of Prodigy SPC the Median & Range chart offers no advantage over the Average & Range chart. Within manual SPC procedures it has the advantage of requiring less mathematics when calculating the points to be plotted. The chart is provided mainly as a means of providing an alternative view of the data that allows the individual points to be compared to the control limits.

The Individual & Range Chart



The Individual & Range chart is an alternative way of viewing the data that is useful in special situations, as the chart is based on individual measurements and not samples. A good use for this chart is where products are made in batches.

Suppose ingredients are mixed together in a large vat and then used to make product. Any variability in the product during this batch run is unlikely to stem from the mixed ingredients, however variability between batches may be a result of inconsistent mixing of ingredients. For this reason you may only want to take readings from the mixing vat once for each batch and monitor the data on an Individual & Range chart.

The Individual & Range chart may also be used to monitor readings that are either expensive or time-consuming to produce and are therefore not produced in sufficient quantities for an Average & Range chart.

The chart should only be used when the frequency distribution of the measurements is known to follow a normal distribution.

The Individual & Range chart is produced as follows.

- The **Ranges** are calculated as being the difference between successive values (hence there will be one less range than there are readings). The sign of the range is always positive.
- The average reading and the average range are calculated. These act as the centre lines for the Individual and Range charts respectively.
- Control limits are calculated for each of the charts.
 - The **Upper Control Limit** for the Range Chart is known as **UCL_r** and is calculated as being $D_4 * (\text{the average range})$.
 D_4 is a statistical constant that varies with the group size.
 - The **Lower Control Limit** for the Range Chart is known as **LCL_r** and is assumed to be 0.
 - The Range Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Range Chart control limits are recalculated. (The out-of-spec points will **not** be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).
 - The **Upper Control Limit** for the Individual Chart is known as **UCL_x** and is calculated as being $(\text{the average}) + 2.66 * (\text{the average range})$.
2.66 is a statistical constant that has been determined for this type of chart.
 - The **Lower Control Limit** for the Individual Chart is known as **LCL_x** and is calculated as being $(\text{the average}) - 2.66 * (\text{the average range})$.
2.66 is a statistical constant that has been determined for this type of chart.
 - The Individual Chart is checked for out-of-spec points and if 2 or less are detected these points are dropped from the data and the Median Chart control limits are recalculated. (The out-of-spec points will **not** be removed from the data if that is the action you have chosen in the **Run Properties** – the data will simply be flagged as having out-of-spec points).
 - Both charts are checked for out-of-spec points and run-rule violations, starting with the Range Chart. If any violations are detected they are indicated on the chart. The process is not in statistical control if any violations occur.

The values plotted on the Individual chart are actual and therefore these values can be compared against the specification and control limits.

When points fall outside of the control limits you should look for assignable causes and take corrective action. When points fall outside of the specification limits but **not** outside of the control limits then the problem is more deep rooted and changes may be required in the plans or the process.

The C_m and C_p values cannot be calculated from data provided by the Individual & Range chart.

SPC In Conjunction With Historical Trends

Prodigy SPC can work in conjunction with historical trends. When viewing an historical trend you can launch an SPC analysis of the data being viewed directly from the Historical Trends toolbar. SPC takes all of the signals that are currently being displayed on the trend and looks for a SPC projects that have been created for those signals. If no project is found for a given signal, SPC creates a new project. If a project is found, SPC then looks at the start and end times of the current historical trend and looks for an SPC run that matches those times. If none is found a new run is created. If an existing run is found, it is selected.

You will finally be presented with an SPC view that contains only projects for the signals that were being viewed in the historical trend. All of the runs for these projects will be available, though, so that you can make comparisons between various runs of the same project.

The Alarm Scheduler

The Purpose of the Alarm Scheduler

It may be a requirement of a particular Prodigy system that some signals have alarm limits that must vary from time to time. For example, a temperature may need to go into alarm at 140°C when one product is being produced and 170°C when another is being produced. The Alarm Scheduler provides a mechanism for defining a series of alarm limits (**Alarm Schedules**) and then applying a given schedule to the signal database when required.

How to Use the Alarm Scheduler

The Alarm Scheduler is a driver and so in order to use Alarm Scheduling you must have this driver loaded. Once this driver is loaded and running, alarm scheduling will become available.

Preparing the Driver

Run Driver Configuration and check if there is already an Alarm Schedule driver loaded. If not, select an empty slot and load the Alarm Scheduler driver into that slot. Set it to **Auto Start** and **Start** it. Exit the **Driver Configuration** dialog.

Do not have more than one Alarm Scheduler driver running – this will lead to undesirable results.

Preparing the Signal Database

As mentioned above, the Alarm Scheduler works by changing the alarm limits for a number of specified signals. It does this by finding the Alarm Strategy associated with each signal in the Schedule and modifying the alarm limits within this Strategy (see The Alarm Strategy Dialog in Signal Database Configuration).

If other signals share this Alarm Strategy then they will all adopt the changes made by the Alarm Schedule. This could lead to undesirable consequences, so when using Alarm Scheduling it is advisable to give every signal a unique Alarm Strategy. As an aid to doing this there is a button on the **Signal Database Configuration** dialog labelled **AutoAssign**.

AutoAssign

This button is only accessible when the Alarm Scheduler is running. Its purpose is to prepare the current signal database for use with Alarm Scheduling by setting each signal to use a unique Alarm Strategy and also setting up these strategies to known initial values.

Ensure that you do not press the AutoAssign button unless you are sure you know what you are doing. If you have spent a long time setting up alarm strategies all that work will be lost when AutoAssign is pressed.

Alarm Schedule Signal

A signal is used to tell the Alarm Scheduler which alarm schedule must be loaded. When an Alarm Schedule is chosen from the **Select Alarms** XForm this signal is set to 1 and the name of the Alarm Schedule that has been selected is written into location 1 of the signal's event text. The Alarm Scheduler detects the change in signal value, reads the Alarm Schedule named in the event text and loads the values specified in this Alarm Schedule into the Alarm Strategy table of the signal database.

The **Select Alarms** XForm requires a signal called **ALARMSIG** to exist in the signal database.

- Create a signal in the Signal Database called **ALARMSIG**.
- Set the signal type to be **Finite State** or **Derived Digital**.
- Give it a range of 0 to at least 1.
- Choose an **Event Text** starting index that is appropriate for the range you have given the signal.
- Select the **Alarm Scheduler** as its driver.

*You do not have to use the **Select Alarms** XForm. Create a signal in a similar manner to that described above and give it a type **Finite State**. Enter the names of your Alarm Schedules as the event text entries for that signal. Use any mechanism available within **Prodigy** to set the value of that signal to the Alarm Strategy required – the Alarm Scheduler will detect the change in value and load the Alarm Schedule named in the event text.*

It is common practice to set a **Default Alarm Schedule** and put its name into the 0 location of the signal's event text. The Default Alarm Schedule is usually used to set the Alarm Strategy values to something that will not produce erroneous or superfluous alarm states when the process is not running, or is in some idle state. The Default Alarm Schedule is then selected by setting the signal value to 0 when required.

Creating Alarm Schedules

Alarm Schedules may be created and modified using a standard XForm **Alarm Schedule**. Each signal within the schedule has its four alarm levels defined and there is also an enabled flag. When the Alarm Scheduler loads the Alarm Schedule then if this flag is ticked, the new alarm strategy is applied to the signal. If it is not ticked then whatever is currently in the Alarm Strategy for that signal will remain unchanged..

Selecting Alarm Schedules

Alarm schedules can be selected and made active by using the XForm **Select Alarms**. Alternatively follow the method described earlier in the Alarm Schedule Signal section and use facilities within Display Builder, for example, to select the required Alarm Schedule.

The Alarm Schedule is overlaid onto the live database, so if **Prodigy** is stopped and restarted the Alarm Schedule will need to be reselected. You can use signal initialisation to have the signal automatically set to the value it had when **Prodigy** was last stopped (see The Recording Strategy Dialog in Signal Database Configuration). Alternatively you can use **S** from within the Additional Programs tab of the Services dialog (see Additional Programs Tab in **Prodigy** Services Control Panel).

Access Database Configuration

Use Of Access Databases

In Prodigy batch and recipe data as well as data from things such as Down Time Logging and SLANG are written into Microsoft Access databases. The Prodigy software is designed so that anything that attempts to open a database that does not exist can automatically create that database, with all the necessary tables, fields and indices. Furthermore, if Prodigy attempts to write a field to a database and that field does not exist, Prodigy can create it “on the fly”.

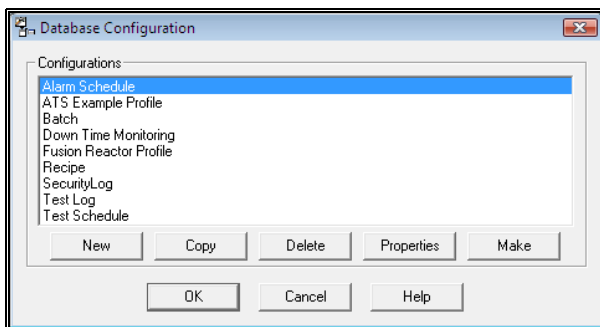
In the past, batch databases were created containing a minimum set of tables, fields and indices, and all other fields were created on the fly by BDR or XForm etc. when they were needed. This proved to be OK for small amounts of data but was found to be inefficient and potentially troublesome for larger data sets, and also provided no means to create further related tables within the database.

In order to provide a more efficient and flexible way of creating Access databases the **Database Configuration** program is now used (do not confuse this with **Signal Database Configuration**).

The Database Configuration Program

Main Dialog

The first dialog of the Database Configuration program is shown below.

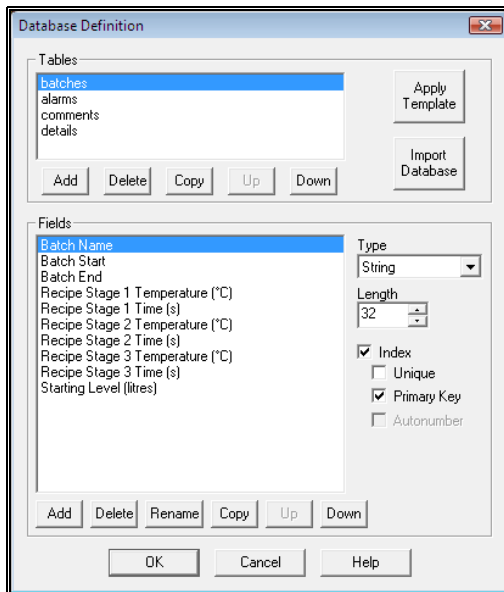


This dialog shows a list of all of the existing database configurations and gives the following options.

- **New/Copy/Delete.**
Allows you to create a new configuration, copy an existing configuration or delete an existing configuration.
- **Make.**
This displays a browser window that allows you to specify the name and location of a database that will be created using the currently selected template. You could then use Access to check that the database is how you expect it to be.
- **Properties.**
Displays the dialog that enables you to specify the tables and fields that will exist in the database (see next topic).

Database Definition Dialog

The database Definition dialog is shown below.

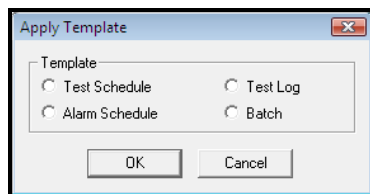


There are two main parts to this dialog – **Tables** and **Fields**.

Tables

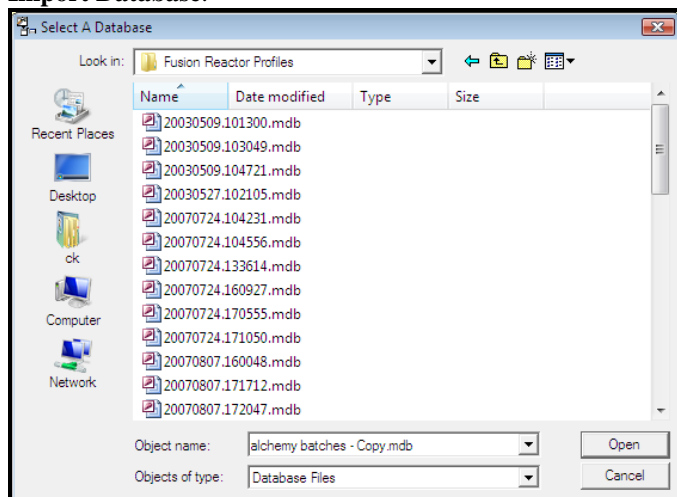
The **Table** section is fairly straightforward.

- **Add.**
Adds a new table to the configuration.
- **Delete.**
Deletes the currently selected table.
- **Copy.**
Copies the currently selected table into a new table with a given name.
- **Up/Down.**
These buttons move the selected table up or down in the list. This affects the order in which tables are created in the database though usually this is irrelevant to the user.
- **Apply Template.**
This button adds a standard template onto an existing or new configuration. After the button is clicked the following dialog is displayed.



This dialog shows the four standard database types that exist within Prodigy. Select the one that the database configuration you are now creating is most like (it is most probably **Batch**) and click OK. The program will then add the tables, fields and indices that are *essential* to the correct use of that type of database. You are then free to add other tables and fields as required, but avoid renaming or deleting anything that was added by the template. The template can be reapplied to a configuration at any time with no ill effects. Sometimes it **must** be reapplied, e.g. when a new table type is added to a configuration (such as the **Attributes** table to the **Batch** template). This will usually be indicated in the release notes.

- **Import Database.**



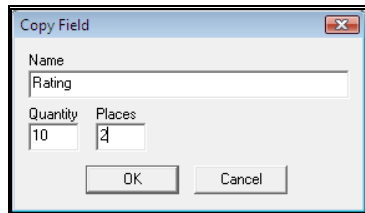
This allows you to read an existing database file and create a configuration from it that will be able to create identically structured databases. If a table name exists in the database that is already used in the configuration it is assumed that the new table will be appended to the existing one (any fields in the new table that are not in the existing one will be added). Any fields that already exist in given tables will not be modified.

Fields

The **Fields** section is a little more involved. As well as specifying the names of the fields in a table, you must specify the data type, the length (for string variables) and also any index parameters.

- **Type.**
The choices here are **double**, **integer**, **boolean**, **string**, **memo** and **date/time**. In general Prodigy only needs three types of variables, **double**, **string** and **date/time**, but the others are added to provide greater flexibility.
Note that if you choose the **boolean** type, Access™ represents this as 0 for false and –1 for true. **Memo** fields are not really recommended – they are basically variable length text fields and are efficient for use of space but inefficient in most other ways. They cannot be used as an index, nor in evaluations and their use in reports is limited. It is much better to use string variables.
- **Length.**
This field is only available for string variable types. Use it to specify the maximum length of the string that will be allowed for this field.
- **Index.**
Specifies that this field is to form part of an index. Indices are created in one of two ways.
 - **Unique.**
If **Index** is checked but **Primary Key** is not checked, the individual field is indexed, to provide secondary sorting capabilities on non-primary fields. The state of the **Unique** check box is used to determine whether this key should be unique or not
 - **Primary.**
If **Primary Key** is selected, then **all** fields that have this option checked are placed together in one primary key (by definition, there can be only one primary key). Primary keys are always unique so the state of the **Unique** check box is irrelevant for primary keys
 - **Autonumber.**
Autonumber fields are a special type of index field. The field type must be Integer and the field is automatically assigned a new unique value when the record is created. The value placed into the field increases by 1 each time a new record is created.
- **Add.**
This adds a new field into the record. You must specify a unique name (within this table) for the field. The field name must not contain period (.), exclamation (!) or square brackets ([or]).

- **Delete.**
Deletes the currently selected record.
- **Up/Down.**
Positions the record within the list which in turn affects the field's position within the table. This is not very important unless you want to view the data using Access and would like the columns in some sort of "sensible" order.
- **Rename.**
Renames the currently selected field.
- **Copy.**
This allows you to create multiple copies of the current field. The following dialog is presented:



Enter the new name of the field to be copied, the quantity of copies to be made, and the number of numeric positions to use for the iterator. The example shown here will result in fields being created called **Rating01, Rating02... Rating09, Rating10**. If any names are created that already exist (say for example **Rating06** already existed) then that number is skipped, but 10 copies are still made and the iterator goes up to 11 instead of 10.

To make another 10 copies, use the name **Rating** again, Quantity 10 and places 2. This will create **Rating11 to Rating20**. **Do not** select Rating01 and make 10 copies if that – you will get **Rating0101 to Rating0110**.

Batch Data Recorder

Purpose Of Batch Data Recorder (BDR).

BDR is a facility that allows users to automatically record batch data to a batch file. It can work in conjunction with or independently of *XForms*. BDR will wait until some start condition occurs and it will then:

- Create the batch.
- Write its start date and time to the batch file.
- Write any specified start data to the batch file.
- Write any specified continuous data to the batch file.
- Reset any specified signals.

BDR then waits until one of a number of end conditions occurs and in the meantime it will, at the intervals specified:

- Write any specified end data to the batch file.
- Write any specified continuous data to the batch file.

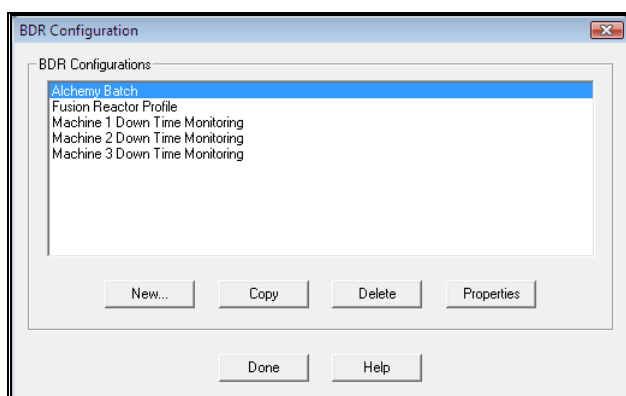
When an end condition occurs (or another start condition) BDR will:

- Write the batch end date and time to the batch file.
- Write any specified end data to the batch file.
- Reset any specified signals.
- Write away any alias tag references.

What determines the start and end conditions, what data is written to the batch file and which signals are reset is determined by *Configurations* and there may be many configurations in use simultaneously. BDR may be being used to record how much product is being produced each hour, each shift, each day, each week and so on. The configurations would specify different start and end criteria based on the time period they are covering.

BDR Configuration Program

The BDR Configuration files are stored in a standard area within the Prodigy directory structure. The BDR Configuration program scans this area and presents a list of all of the configuration files present.



- **New.**
This creates a new configuration. A dialog will prompt for the name of the new configuration, which must be unique.
- **Copy.**
Selecting an existing configuration and pressing the Copy button makes copies of existing configurations. A dialog will prompt for the name of the new configuration, which must be unique. The existing configuration will be copied to the new name. When either a **New** or **Copied**

configuration is created, the configuration file is written away with the **Active** flag set to false. This is so BDR will not try to process the configuration until the user has finished working on it.

- **Delete.**
Existing configurations can be deleted by selecting them in the list and pressing the **Delete** button. A confirmation is requested before deleting the configuration.
- **Properties.**
By double clicking on the required selection (or by single clicking and then pressing the **Properties** button) the user is presented with a tabbed dialog that allows them to set or alter aspects of the currently selected configuration.

Configuration Properties

The configuration screen for BDR has several tabs. These are described in detail below.

Main

The screenshot shows the 'BDR Configuration - Alchemy Batch' dialog box with the 'Main' tab selected. The dialog is divided into three main sections: 'Run Parameters', 'Start Condition', and 'End Condition'.
Run Parameters: Includes a checked 'Active' checkbox, an unchecked 'Separate Databases' checkbox, and text boxes for 'Batch Database' (pointing to a directory), 'Database Config. File' (pointing to a file), 'Batchname Specification' (containing '<\$BATCHID>'), and 'Batchname Signal' (containing 'BATCH'). There are 'Browse' and 'Select' buttons for these fields, and a 'Reports...' section with 'Select Report' and 'Select Crystal' buttons.
Start Condition: Includes an unchecked 'Overwrite when end date exists' checkbox, a text box for 'Overwrite when start date in existing file is older than' (set to '0s'), and three options: 'Signal changed' (with 'START' selected), 'Batch name changed', and 'Expression' (checked, with '\$ALCRUN = ON AND \$BATCHID = 1' and 'is True' selected). A 'Regularly' option is also present with 'Every 1m starting at 0:00'.
End Condition: Includes two unchecked options: 'Signal Changed' and 'Batch name changed', and 'Expression' (checked, with '\$ALCRUN = OFF' and 'is True' selected). A 'Regularly' option is also present with 'Every 1m starting at 0:00'.
At the bottom are 'Save', 'Done', and 'Help' buttons.

The data on this tab is grouped into three sections, **Run Parameters**, **Start Condition** and **End Condition**.

Run Parameters.

- **Active.**
If this box is checked, this BDR configuration will be run by BDR when **OK** is pressed. If the box is not checked, the configuration will not be run, but if it is already running it will be stopped. This mechanism is used to determine which configurations should be running and which should not.
- **Separate Databases.**
Check this box if each batch is to be placed in its own database. The database will have the same name as the batch and in this instance only the path of the directory in which the databases are created is required in **Batch Database** above.
- **Batch Database.**
This specifies the name of the database in which the batches created by this configuration will be stored. If **Separate Databases** has been specified only the directory path is required, not the file path. Use the **Browse...** button to locate the correct database or directory.

- Database Config. File.**
This is the name of the Database Configuration file that is used to create an empty database file if the database does not already exist. This is the most efficient and failsafe way to create empty databases with all of the correct fields and tables required for BDR, XForms or reporting.
- Batchname Specification.**
This field allows the user to define how the batch name is formed. The field can consist of freeform text interspersed with any number of special fields, detailed below. The batch name is formed by substituting any special fields in the string with their actual values, these values being recalculated at regular intervals. If, when one of these variables is recalculated it causes the batch name to change, this will trigger any start or end condition that depends on change of batch name. Special variables are contained in angle brackets < > and can be one of the following:

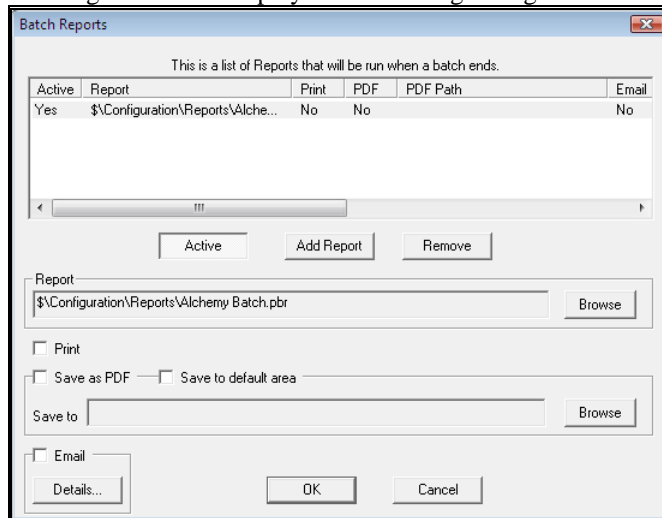
 - <\$TAGNAME>.**
A signal database tagname may be entered, e.g. <FLOWRATE>. The value of FLOWRATE will be substituted in the string unless the tag is that of a *finite state* or *digital* signal, in which case the relevant *event text* entry will be substituted.
 - <YEAR>.**
The current year, shown as four digits, will be substituted.
 - <MONTH>.**
The current month, shown as two digits, will be substituted.
 - <DAY>.**
The current day, shown as two digits, will be substituted.
 - <HOUR>.**
The current hour, shown as two digits, will be substituted.
 - <MINUTE>.**
The current minute, shown as two digits, will be substituted.
 - <SECOND>.**
The current second, shown as two digits, will be substituted.

Example a Batchname Specification string of
BATCH-<\$GYLE>-<YEAR><MONTH><DAY><HOUR> would give a batch name of **BATCH-23-1998011610**, assuming it is 16th January 1998, somewhere between 10am and 10:59am and the value of the integer signal **GYLE** is 23. If the start condition had been specified as *Batch name Changed* (see later) a new batch will be started every hour, so at 11 O'clock the batch name will change to **BATCH-23-1998011611**, and so on
- Batchname Signal.**
Select the name of the signal that will be used to communicate the current batch status and name. When a new batch begins, BDR will set this signal to 1 to indicate that a batch is running and it will set the event text entry for position 1 of this signal to be the name of the current batch. This enables Display Builder objects to be configured to display the current batch status.
- Reports.**
This allows you to define a list of *Prodigy Reports* or *Crystal Reports* that will be run when the batch terminates (see the relevant chapters on how to create various reports within Prodigy).

Reports

- **Select Report.**

Clicking this button displays the following dialog:



- **Active.**

Reports may be temporarily disabled or enabled by using the **Active** button. Only **active** reports are run when a batch completes.

- **Add Report.**

Use this button to add a report. This will display a standard browser that will allow you to choose the required report file.



- **Remove.**

Use the **Remove** button to remove any reports that are no longer required.

- **Browse.**

Use this button to browse for a new (or different) report.

- **Print.**

Check this box if the report is to be automatically printed.

- **Save as PDF.**

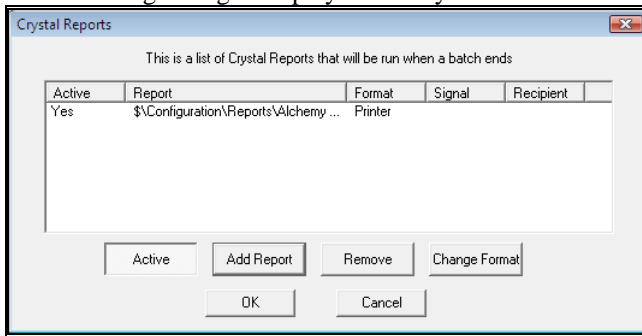
This saves the report (after it is run) as a PDF file (requires PDF Output license). The output will have the same name as the batch but will have the extension .pdf.

- **Save to default area.**

If this is checked the PDF output goes to the default area specified on the **Prodigy Control Panel**.

- **Select Crystal.**

The following dialog is displayed when you click the *Select Crystal* button.



- **Add Report.**

Use this to select reports to be added.

- **Active.**

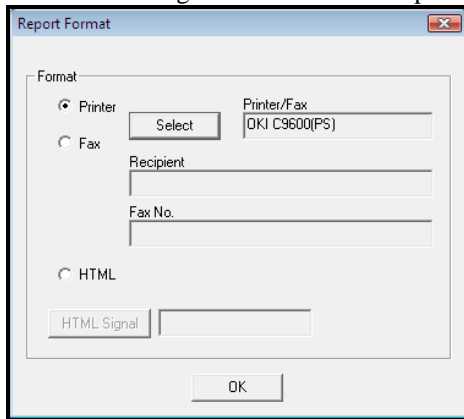
Use this to make these reports active or inactive (only active reports will be run).

- **Remove.**

Use this to remove a report that is no longer required.

- **Change Format.**

Use this to change the format of the report output (see below).



This selects the format of the output of the report. The choices are *Printer (which includes Fax)* or *HTML*.

- **Printer.**

If this is selected the report will be output to the printer that is selected. Note that the printer must be available to the User ID that the Prodigy Services are logged as.

- **HTML.**

If this is selected the report will be output in HTML format to a file. This file may be viewed in any web browser. The path (including the name) of the file can be specified via a signal. Click the *HTML Signal* button to display the SigScan dialog and select an appropriate signal (it should be a derived digital). Before the report is run, this signal should have the path and name of the output file placed into its event text 1 location, and the signal value should be set to 1. If no signal is given or if the signal value is not 1, the report will be output to C:\Temp\CrystalHTML.htm

Start Condition

This specifies when a new batch is to be started and what actions to take if the batch name provided (or derived) already exist.

- **Overwrite when end date exists.**

If this option is checked and the batch already exists but it has an end date (i.e. the batch has been completed) then delete the existing batch and begin a new one with the same name.

- **Overwrite when start date in existing file is older than.**

If the start date in the existing batch is older than the time period specified here, then delete the

existing batch and begin a new one with the same name.

*If a batch exists and it is not overwritten (because neither of the options specified above satisfies the condition that it **should** be overwritten) then the existing batch is used and is added to. When the batch ends a new end date will be written.*

- **Start Condition.**

This is the part of **Start** that specifies when a new batch is to be created.

- **Signal Changed.**

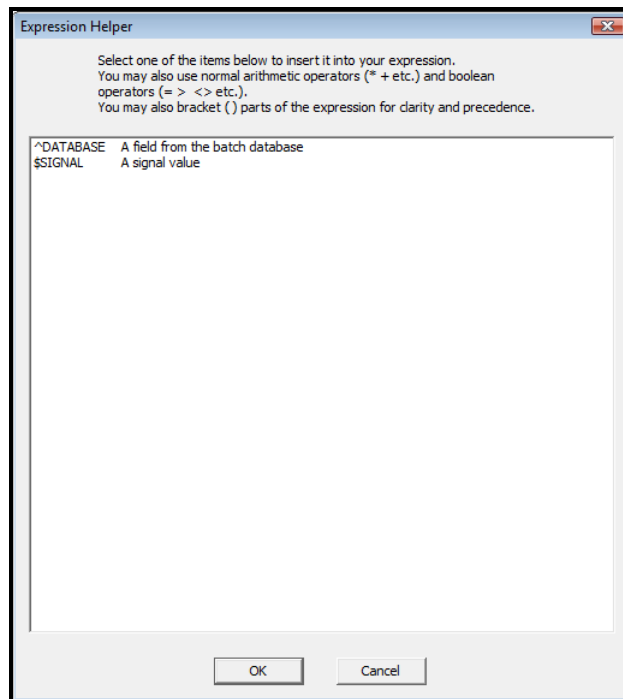
A new batch will be started when the value of the signal specified here changes. For instance, this could be a digital signal that specifies whether the process is running or stopped. Use the **Select...** button to invoke the SigScan dialog to choose the required signal.

- **Batch name Changed.**

If the batch name (as specified in Batchname Specification, above) changes, a new batch will be started.

- **Expression.**

If the value of an expression entered here evaluates to **true**, i.e. non-zero, a new batch will be started. Expressions are as for **Evaluated Signals** and the expressions available for use can be seen by pressing the **Select** button. Pressing the select button will bring up the following **Expression Helper** dialog.



In this case only databases and signals are available for use though there are many other functions available. The Expression Helper in End Data has the ability to make use of all the functions this facility offers so full details of the Expression Helper are available in the end data section of this chapter. The **Test** button to the right of the **Select** button can be used to test the expression.

- **Regularly.**

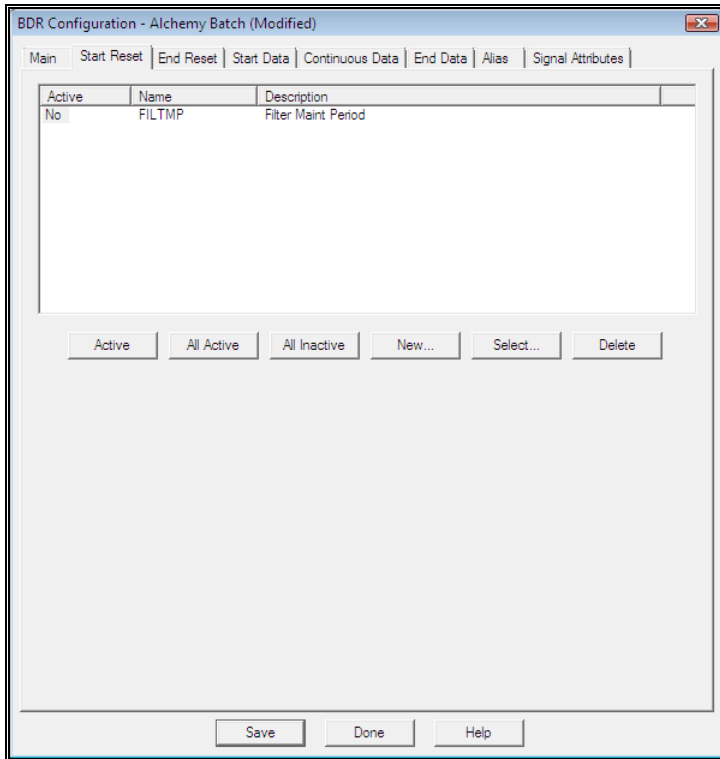
A new batch is started at some regular interval. Maybe a new batch is started every hour, or at twenty past every hour. Specify exactly when the new batch is started is by clicking the **Select...** button.

End Condition

The options here are exactly as for **Start Condition** except that it is possible to not have any end conditions. If this is the case, a batch does not end until a new batch is begun. This is most likely the case in situations where a process runs continuously but new batches are begun every so often, e.g. every hour. End conditions

are most likely to be defined when a process runs for the duration of a batch and is then stopped for some period of time before a new batch is begun.

Start Reset



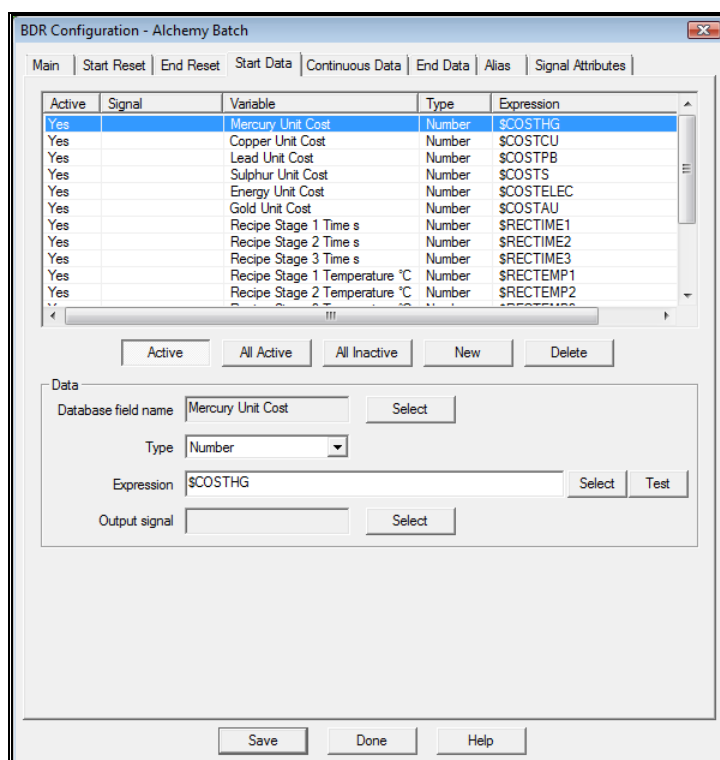
This tab provides a list of signals that are to be reset (i.e. set to zero) every time a batch is started. When a new batch is started BDR resets every signal in this list whose **Active** flag is set to Yes. The buttons on this tab are described below.

- **Active.**
This toggles the active flag for the currently selected signal. This mechanism is provided in order to aid the user in designing and testing strategies. When a final strategy has been decided upon, it is likely that all signals that do not have their active flag set will be deleted from this list.
- **All Active.**
Activates all signals in the list.
- **All Inactive.**
Deactivates all signals in the list.
- **New...**
Allows new signals to be entered into the list. The SigScan dialog will be displayed.
- **Select...**
Allows a different signal to be selected to replace the currently selected signal. The SigScan dialog will be displayed.
- **Delete.**
Deletes the currently selected signal from the list.

End Reset

This is exactly as for the Start Reset tab except that it specifies signals that will be reset when a batch ends.

Start Data



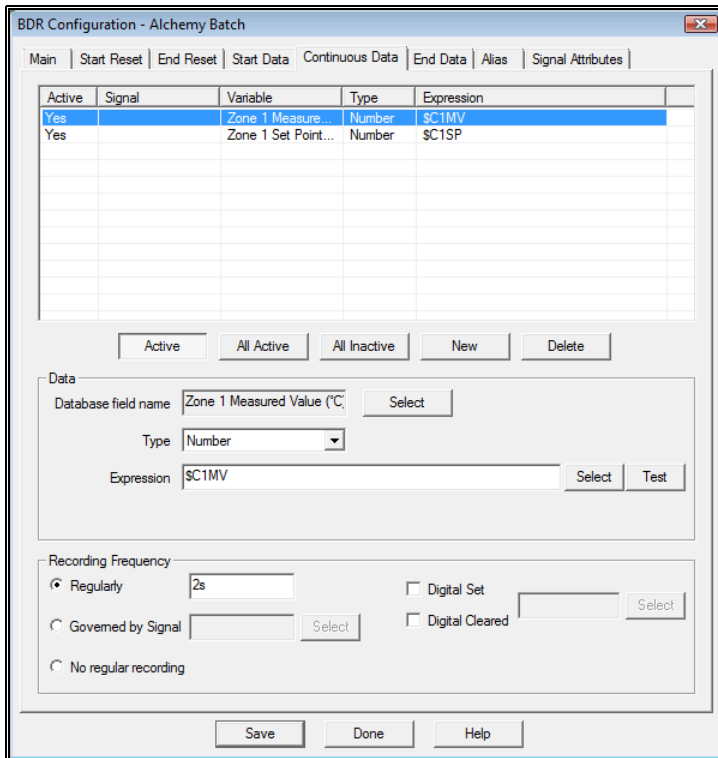
This tab provides a list of batch file variables that will be written when a new batch is started. When a new batch is started BDR writes every variable in this list whose *Active* flag is set to *Yes*. The fields and buttons on this tab are described below.

- **Active.**
This toggles the active flag for the currently selected variable. This mechanism is provided in order to aid the user in designing and testing strategies. When a final strategy has been decided upon, it is likely that all variables that do not have their active flag set will be deleted from this list.
- **All Active.**
Activates all variables in the list.
- **All Inactive.**
Deactivates all variables in the list.
- **New...**
Allows new variables to be entered into the list.
- **Delete.**
Deletes the currently selected variable from the list.
- **Database Field Name.**
This is the name of the variable, as it will appear in the batch file.
- **Type.**
This is the type of the variable that will be written in. Five types are allowed
 - **Event Text**, this will take the event text associated with an appropriate signal
 - **Number**, which is numeric data.
 - **Date/Time**, which is used to represent dates.
 - **Fixed Text**, a fixed entry of text that will be the same every time a batch is created
 - **String**, which is text data.
- **Expression.**
This is an expression, as briefly described earlier, whose evaluated value will be written to the batch file. The **Test** button can be used to test the validity of the expression.

- **Output Signal**

The value can optionally be written to a signal. If this is required, enter a signal name here by selecting it with the **Select...** button. If a batch is restarted, the existing data is read from the batch database and used to initialise this signal to the state it had when the batch ended.

Continuous Data



Continuous data is exactly like Start and End data except that it is written into a different table in the batch database and it is *timestamped*. The data is written repeatedly, at an interval dictated by the configuration, but unlike *End Data* it does not overwrite the previous value, it writes a new record with a new timestamp.

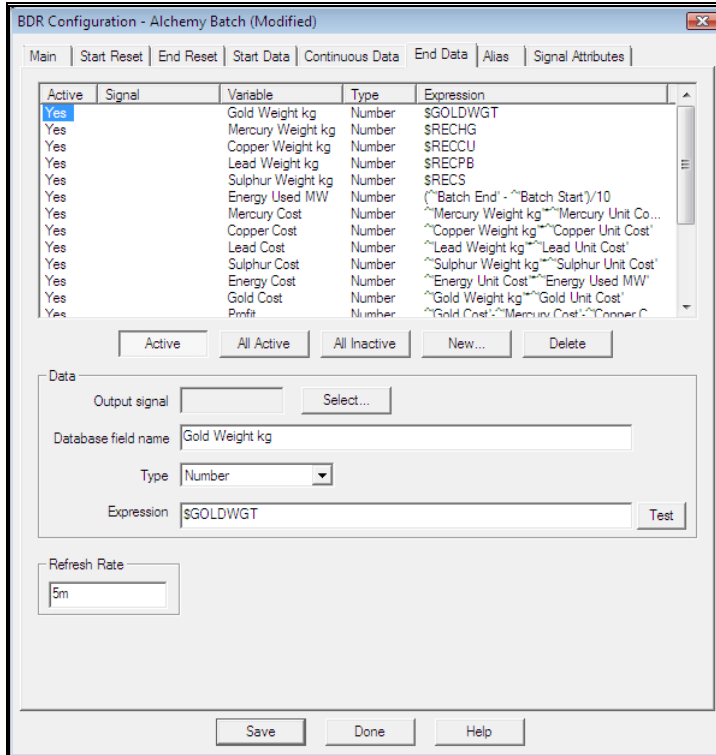
The main fields are exactly as for *Start Data* but there are additional fields at the bottom of the dialog which govern the frequency of the recordings.

- **Regularly.**
If this is checked then the data is written at the time interval specified to the right of this.
- **Governed By Signal.**
If this is checked then the time interval is determined by the value of the signal specified to the right of this. The signal should be set to a value that represents the time interval in seconds or fractions of seconds. This time interval can be modified “on the fly”, maybe by a SLANG program. Any changes to the value of the signal will have an immediate effect. If the signal is set to zero, recording of continuous data will be suspended until the value is made non zero again.
- **No Regular Recording.**
Check this option if there is to be no *regular* recording of the data. The data can still be recorded, by one of the manual methods mentioned next.
- **Digital Set.**
When this option is checked the data is written when the signal specified to the right of this field is set (i.e. is non zero). This option may be used in addition to the regular recording options and provides a means of making additional manual type recordings. When the recordings have been made the signal value is set back to 0.
- **Digital Cleared.**
This is exactly as for *Digital Set* except that the recordings are made when the signal value is set to zero. After the recordings have been made the signal value is set to 1.

In addition to all of the above, the data is written immediately that the batch starts and again when it ends.

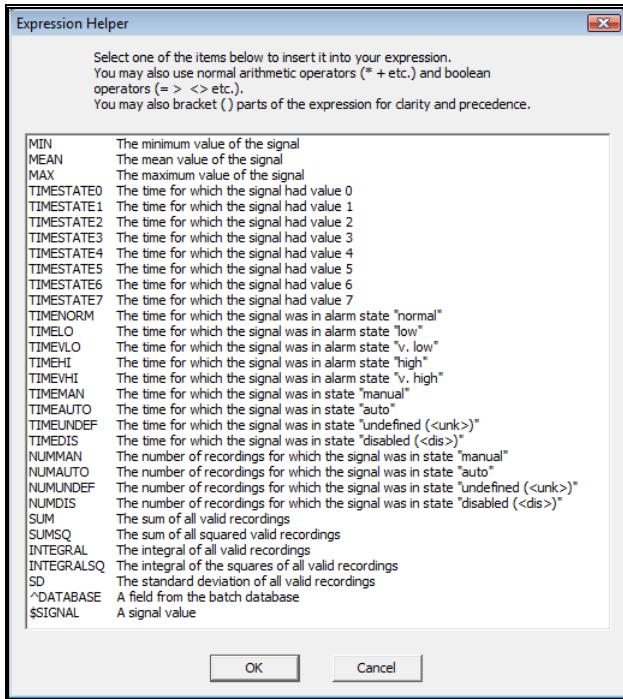
Records that are triggered manually (by the setting or clearing of a digital) will have the label **Manual** written into the **Type** field of the record. Automatic records have the label **Auto** written into that field.

End Data



This is as for the **Start Data** tab except that the variables are written to the batch file when a batch ends. In addition, these variables are written at an interval determined by the **Refresh Rate**. This determines the frequency with which the **End Data** is rewritten. The value entered should be in the standard Prodigy time period format. In our example **5m** means five minutes and would have the effect that every five minutes after the batch start date the end data would be written to the batch file. Each time they are written they simply overwrite the previous values, unlike the **Continuous Data**.

More manipulation of the data can be achieved through the Expression field as there are more functions available in the expression helper. The expression helper dialog can be seen below with all of its available functions.



Descriptions of the statistical functions available are given in the dialog, but below is a brief description of what each function does:

*BDR expressions can make use of **statistical operators**. These are of the form <Operator> <Signal> (e.g. MAX \$FLOWRATE) and are defined as follows:*

MEAN, MIN, MAX: *These give the mean, minimum and maximum value of the specified signal over the period of the current batch.*

TIMESTATE0, TIMESTATE1, TIMESTATE2, TIMESTATE3, TIMESTATE4, TIMESTATE5, TIMESTATE6, TIMESTATE7: *This is the time that a finite state or digital signal has the value 0, 1, 2, 3, 4, 5, 6 or 7 (only 0 or 1 with digital). If used with real or integer signals then states 0, 1, 2, 5 and 6 give the time that the signal was in the alarm state **Normal, High, Very High, Low** or **Very Low** respectively. In other words, they are interchangeable with the functions given below.*

TIMENORM, TIMELO, TIMEVLO, TIMEHI, TIMEVHI: *These return the amount of time that a signal has been in one of the specified alarm states, Normal (i.e. not in alarm), Low, Very Low, High or Very High, respectively.*

TIMEMAN, TIMEAUTO, TIMEUNDEF, and TIMEDIS: *These return the time that a signal value has been in the relevant state Manual, Auto, Undefined or Disabled. (A Manual state means that an operator entered the signal value in some way, via a Display or XForm perhaps. An Auto state means that the signal value came from some piece of plant equipment. An undefined value means the signal value was not set by any means and was therefore unknown. Disabled means that the signal has been disabled so that its value cannot be set by any means.)*

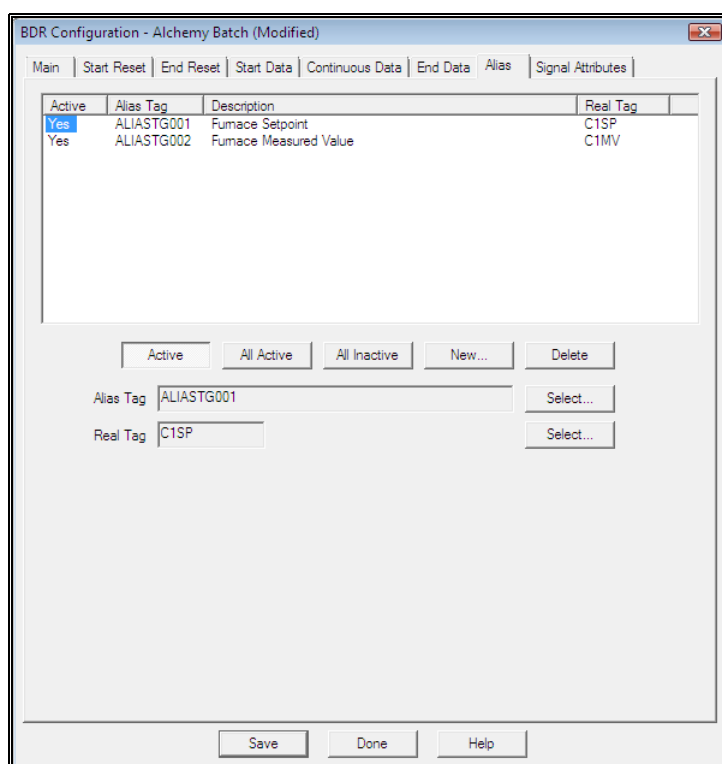
NUMMAN, NUMAUTO, NUMUNDEF, and NUMDIS: *This is as above, but returns the **number** of measurements of the specified type, rather than the **time**.*

SUM, SUMSQ: *The sum of all enabled measurements or the sum of all **squared** enabled measurements.*

INTEGRAL, INTEGRALSQ: *Integral of all enabled measurements with respect to time over enabled time or integral of **squared** enabled measurements with respect to time over enabled time. In other words it calculates the area under the curve of the signal values. If the signal were measuring flow rate in a pipe, the integral would give the amount of substance that had actually flowed through the pipe over the duration of the batch.*

SD: *The standard deviation of signal measurements.*

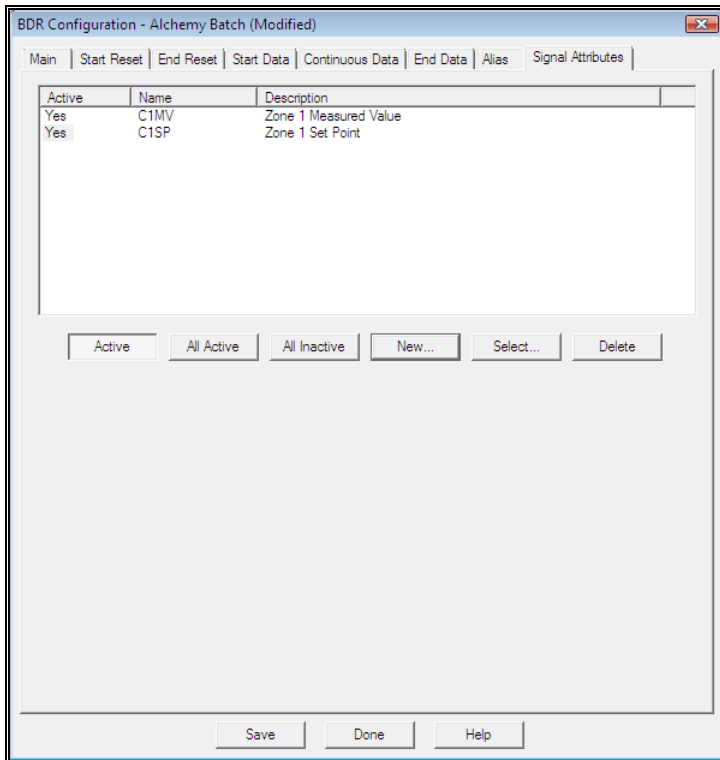
Alias



Alias Tags are used to allow different **real** tags (signals) to be substituted into trends depending on what batch has been run. This is achieved by writing the alias tag and it's equivalent real tag into the batch record when BDR completes. This tab of the BDR configuration screen allows you to specify the alias tag to real tag relationship for batches created by this BDR configuration. (Remember, BDR may be handling several configurations at once, monitoring different batches in different parts of the plant). See the section of the manual dealing with the creation of Alias Tags for more details.

- **Active.**
Press this to make the current selected row active or inactive. Only active rows have data written to the batch record.
- **All Active.**
This sets all rows active.
- **All Inactive.**
This sets all rows inactive.
- **New.**
This adds a new row. Use the **Select** buttons to choose the relevant alias and real tags.
- **Delete.**
Deletes the currently selected row.
- **Select (Alias Tag).**
Allows you to select the alias tag for this data row. It displays the standard Alias Tags dialog.
- **Select (Real Tag).**
Allows you to select the real tag for this data row. It displays the standard SigScan dialog.

Signal Attributes



Sometimes a system may be required to use the same inputs for different types of readings. For instance at one time AI01 may be a temperature input while at another time it is a pressure input, depending on what type of device is connected to the system at the time of use. While a particular set of equipment is connected the signal database may be modified so that the descriptions show correct and meaningful values. This modification will probably be made via an XForm and/or SLANG. BDR can be configured so that it will take the attributes of the signal at the time that the batch is created and record them with the batch. When this batch is subsequently graphed in either a **Historical Trend** or a **Proof Of Process** report, the attributes from the batch can be used in place of the attributes currently assigned to the signal.

To add signals to the Signal Attribute recording section simply click the **New...** button and select a signal. Set it to **Active** to record it, or **Inactive** to temporarily exclude it. The **Select** button allows you to select a different signal for the current row (in effect it's a rename) and the **Delete** button deletes the currently selected row. **All Active** and **All Inactive** will make all rows either active or inactive respectively.

The attributes that are recorded to the batch are

- Description
- Units
- Range Low
- Range High
- Alarm Very Low
- Alarm Low
- Alarm High
- Alarm Very High

Important:

In order to use this facility the batch file **must** include the **Attributes** table. To add the Attributes table to your batch files you must use a **Database Config File** that is based on the Batch template (see **Main**, above and also the chapter on **Access Database Configuration**). In addition, any existing Access Database Configurations that are used by BDR must have the Batch template reapplied to them to ensure that the

Attributes table is added. If Prodigy is already running when the Access Database Configuration is modified you must stop it and restart it so that the new table can be added to the batch files.

BDR in Action

- BDR Scans the BDR area for all configuration files (files with the suffix .bdr). Each file that it finds is read in and parsed to check for errors. Some errors are fatal, some are non-fatal. All errors are reported to the Internal Error Manager and in the case of fatal errors, the configuration is ignored.
- If the Active flag for the configuration is set, the configuration will be processed.
- The Start Conditions are tested. If a test proves positive a new batch is started. The batch name is evaluated from the Batchname Specification field and the database specified in the Batch Database field is opened. BDR checks to see if the given batch name already exists. If the batch does exist BDR determines if it is to overwrite this batch with a new one (i.e. write a new batch start date and delete any data that has been written to this batch) or to continue the existing batch. If a new batch is created, all signals in the Start Reset list that are active are reset to zero and all variables in the *Start Data* and *Continuous Data* lists that are active are written to the batch database. If the batch is continued, and it is the first time that a batch has been run since Prodigy was restarted, any data in the *Start Data* list that is copied to a signal value is read from the batch record and into the copy-to signals.
- BDR cycles, checking on each cycle to see if either the start condition or the end condition for the configuration is satisfied. BDR will only be checking to see if either the start or the end condition is satisfied, but not both. If the user has specified an end condition then BDR will only be checking to see when this condition has been satisfied. When no end condition is specified, BDR checks to see when a new start condition is satisfied.
- While BDR is cycling, it will write the *End Data* (specified in the End Data list) to the batch file every time it has been cycling for the time period specified in the Refresh Rate field. It will also write *Continuous Data* with a new timestamp if the interval for writing such data has occurred, or if the manual writing criteria have been satisfied.
- While BDR is cycling it is constantly checking to see if the configuration file has been modified. If it detects changes, the new file is read and parsed and any changes made may have an immediate effect. If, for example, the end condition is changed, BDR will now be waiting for the revised condition to be satisfied. If the Active flag is altered, BDR will either start or stop running the configuration accordingly. If any changes are made to the End Data list, the variables will be written to the batch file the next time that the Refresh Rate cycle is satisfied.

BDR in Conjunction with XForm

XForm may be used to create batch files, write data to them and start and stop batches. It is simple to use BDR in conjunction with XForm by specifying suitable start and end criteria. For example, the XForm may be made to write the batch name to the event text of a digital signal (e.g. BATCH) and to set this signal's value to 1 when the batch is started. Enter this signal name in the **Batchname Specification** and specify a start criteria of **Expression**, giving the expression as **\$BATCH=1**. BDR will detect that XForm has started a new batch and will pick up the name of it from the event text entry of BATCH. Make sure that the **Overwrite when start date in existing file is older than** field is sufficiently large so BDR will not delete the data in the batch created by XForm and start a new one!

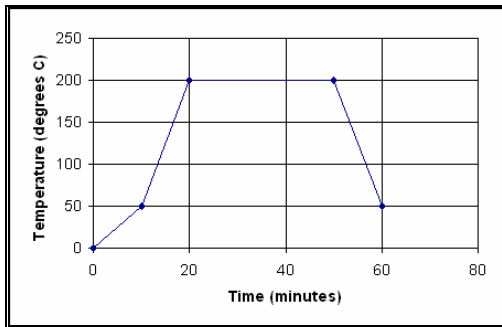
An end condition can be specified that uses **\$BATCH=0** if an XForm is to be used to specify that the batch has ended.

Automated Test Sequencer (ATS)

What Is ATS?

ATS is intended to aid in automating the testing process, allowing the same sequence of events to be reproduced every time. A test is made up of many parts (phases), and each run through of these phases is known as a cycle.

For example:-



The above test consists of 4 phases:-

Phase 1: raise the temperature from 0 to 50 degrees C over a period of 10 minutes (the gradual change of a value by ATS during a phase is known as a ramp).

Phase 2: ramp the temperature from 50 to 200 degrees over a period of 10 minutes.

Phase 3: hold the temperature at 200 degrees for 30 minutes.

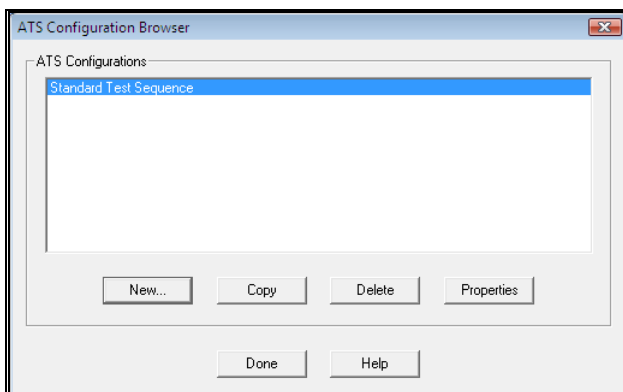
Phase 4: ramp the temperature from 200 to 50 degrees over a period of 10 minutes.

All 4 of these phases carried out sequentially go to make up the test cycle.

Configuration

The configuration of ATS is reached from the Prodigy Control Panel.

Configuration Browser



Use this screen to create a new configuration, copy an existing one, delete an existing one or to access the properties of an existing configuration.

Click Properties to get to the configuration...

Global Options

ATS Configuration

Global Options | Signals | Values | Ramps | Triggers | Counters | Alarm Schedules | End Conditions | SLANG

*Configuration name: Active

Database

*Test schedules dir/file:

*Database Type:

*Database Table:

*Target Phase Length: Signal:

*Cycle time: seconds

* denotes required, all other fields are optional

- **Configuration name.**
The name of your configuration as used in the Configuration Browser
- **Active.**
Tick to have Prodigy automatically start running this configuration when it starts
- **Test schedules directory.**
The directory where you keep the test schedules you will use
- **Database type.**
The type of database you will use for your test schedules, as defined with the Database Configuration tool
- **Database table.**
The name of the table within the database that contains the data fields to be used in the test schedule
- **Target Phase Length.**
The database field that contains the maximum length of time a phase may last
- **Signal.**
The signal into which the target phase length for the current phase will be set
- **Cycle time.**
How often ATS will attempt to work out end conditions and set new values for ramps

Signals

ATS Configuration

Global Options | Signals | Values | Ramps | Triggers | Counters | Alarm Schedules | End Conditions | SLANG

*Schedule ID:

Control

*Start flag:

Pause flag:

*Abort flag:

Start

Start cycle number:

Total cycles:

Start phase number:

Start duration time:

Status

Duration time:

Time to end of test:

Cycle number:

Phase number:

Phase time:

Time to end of phase:

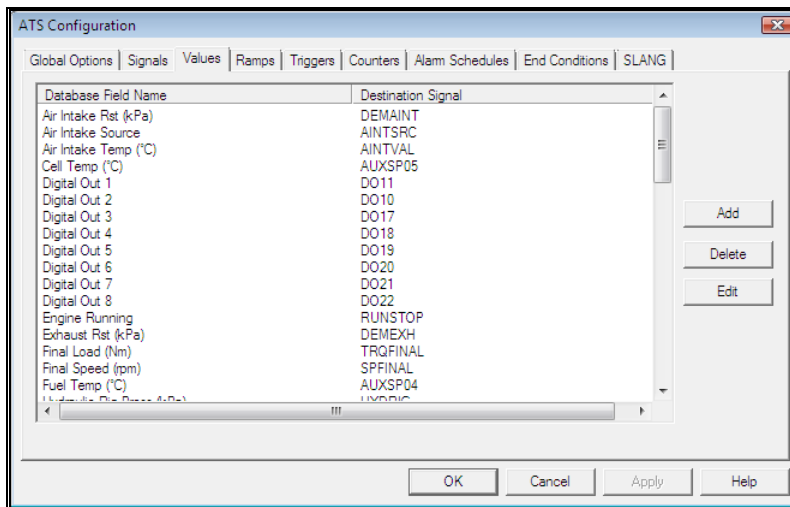
Final Phase:

* denotes required, all other signals are optional

- **Schedule ID.**
The signal into which the name of the schedule will be set prior to starting the test

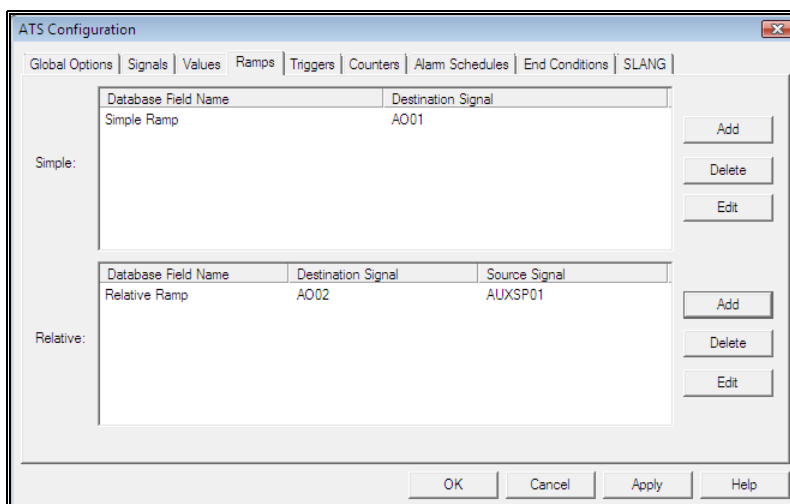
- **Control.**
The following signals are used to control the test
 - **Start flag.**
The signal which will be set to 1 to indicate that ATS should start the test. N.B. ATS will set this signal to 0 when the test ends
 - **Pause flag.**
The signal which will be set to 1 to indicate that ATS should pause the test
 - **Abort flag.**
The signal which will be set to 1 to indicate that ATS should abort the test
- **Start.**
The following signals are used to set various starting values for ATS
 - **Start cycle number.**
The number of the cycle that ATS should start from
 - **Total cycles.**
The maximum number of cycles that ATS should repeat
 - **Start phase number.**
The number of the phase that ATS should start from. If this is changed during a test sequence, the test will advance to the newly set phase number.
 - **Start duration time.**
The duration time that ATS should use to start from
- **Status.**
The following signals are set by ATS in order to show the status of the test
 - **Duration time:**
How long the test has been running
 - **Time to end of test.**
How long until the end of the test (calculated by adding up the target phase times)
 - **Cycle number.**
The current cycle number
 - **Phase number.**
The current phase number
 - **Phase time.**
How long into the current phase the test is
 - **Time to end of phase.**
How long until the end of the current phase (based on the target phase time)
 - **Final Phase.**
The number of the last phase in the test

Values

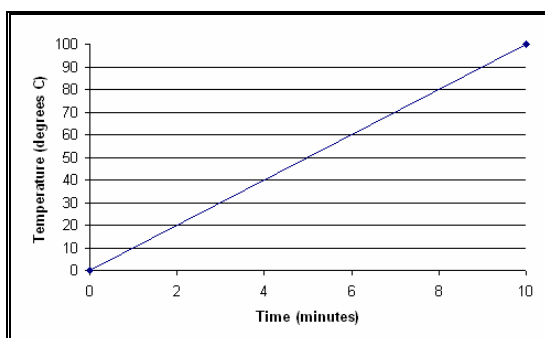


Values are read from the test schedule database (using the Database Field Name) and set in the Signal Database (using the Destination Signal) at the beginning of a phase.

Ramps



Ramps are signals (as specified in Destination Signal) which are moved from their value at the start of the phase to the value specified by the Database Field Name in the test schedule. The values are calculated based on the Target Phase Time and the difference between the start and end values, so when plotted on a trend (see below) the signal's value would show as a straight line between the start and end of phase. The rate at which new values are calculated (the "smoothness" of the line over time) is determined by the Cycle Time set in Global Options.



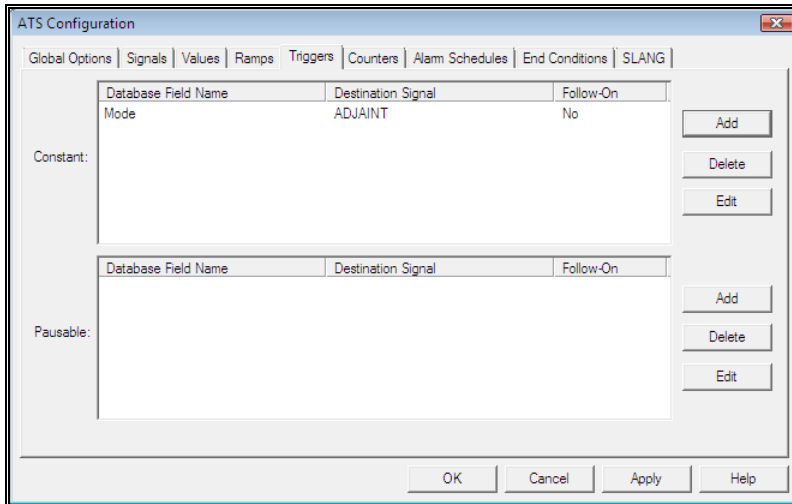
E.g. A phase that lasts 10 minutes and ramps a signal value from 0 to 100 degrees C over the phase.

There are two types of Ramp.

- A Simple Ramp (the Destination Signal is ramped from whatever it is set at the start of the phase to the value supplied in the test schedule).
- A Relative Ramp (the Destination Signal is ramped from the value of the Source Signal to the value supplied in the test schedule).

Setting of Values always happens before Ramps, so the two may be combined to set a Signal Value and then ramp it to a final value over the duration of a phase.

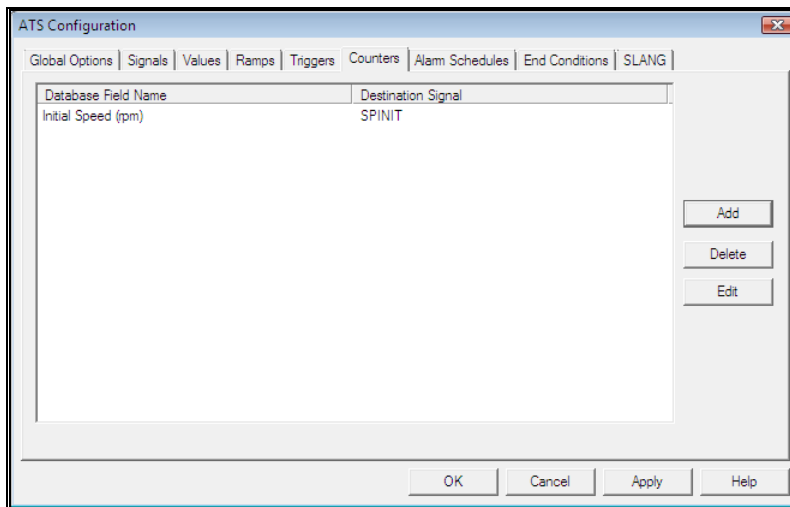
Triggers



Triggers are Signals (as specified by the Destination Signal) that are set to the value 1 or TRUE at a regular time interval (as determined by the Database Field Name in the test schedule). E.g. a value of 80 for Trigger 1 in the configuration above would cause signal TRIGGER1 to get set to 1 every 80 seconds. Constant Triggers are set every specified period, regardless of whether the test is paused or running. Pausable Triggers are paused while the test is paused i.e. an 80 second trigger would be set after 90 seconds if the test was paused for 10 seconds.

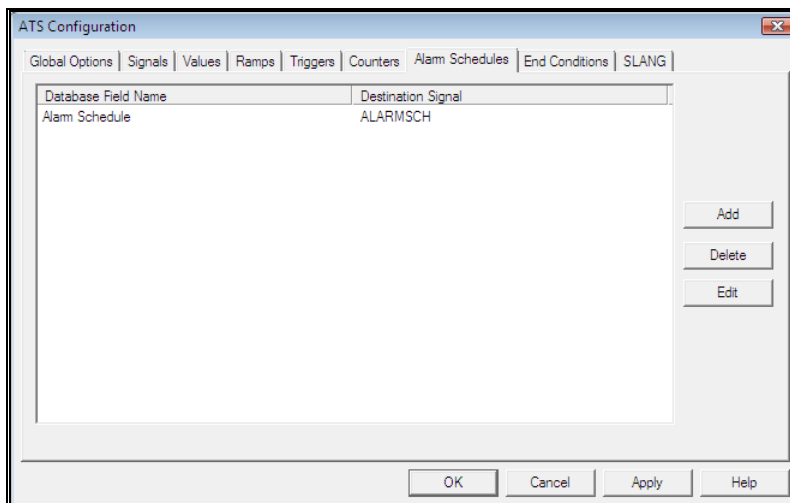
Follow-On behaviour is designed for Trigger's that are required to operate across phases. E.g. If you had a Trigger that you wanted to be set every 10 minutes, but your phases were shorter than 10 minutes, setting Follow-On behaviour would cause the time accumulated at the end of the phase to be carried over into the next phase. (Normally the time associated with a trigger is reset at the start of a phase). In order for this behaviour to occur, the time for the trigger must also be the same across the phases. E.g. If there were two phases, each lasting 5 minutes, and the test schedule had a value of 8 minutes for a Trigger set for both phases and Follow-On behaviour set, the Trigger would be set 3 minutes into phase 2.

Counters



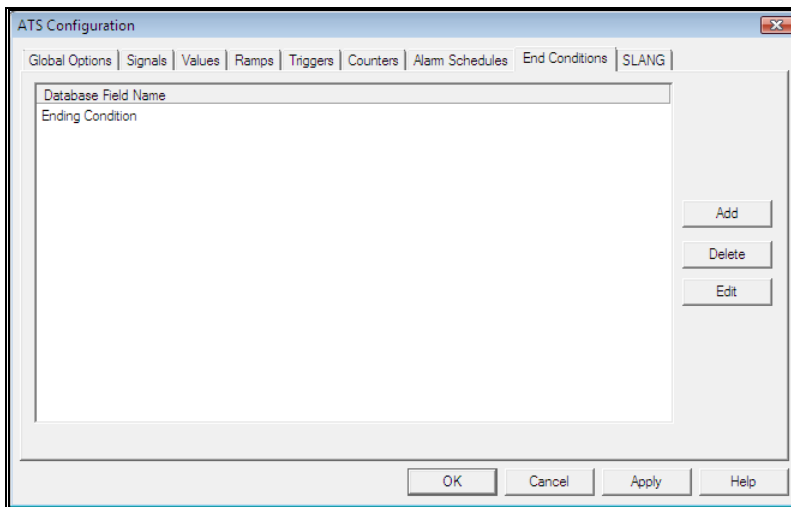
Counters are Signals (as specified by the Destination Signal) whose value is incremented/decremented by the amount specified in the test schedule (as specified by the Database Field Name) at the start of a phase.

Alarm Schedules



This enables ATS to set Prodigy Alarm Schedules at the start of a phase. The name of the required Alarm Schedule is loaded into the Destination Signal from the Database Field specified in the test schedule.

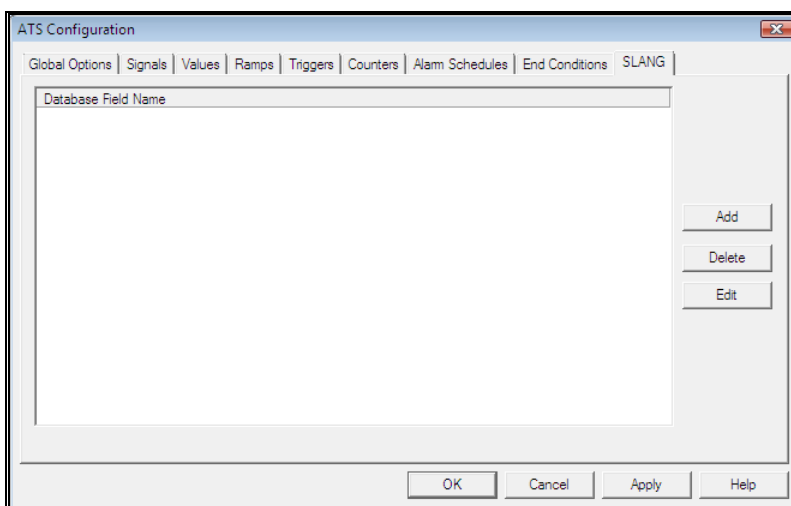
End Conditions



An End Condition is an evaluated expression (e.g. \$TEMP01 > 50) that is calculated every Cycle Time.

If the End Condition is true, the phase will be forced to end before the Target Phase Length has been reached. The End Condition is held in a test schedule field as specified by the Database Field Name.

SLANG



SLANG programs may be started at the beginning of a phase. The programs to be run are specified in the test schedule in the Database Field Names listed here. The test schedule field may contain single program names, or may have many programs listed, each separated by a comma.

Down Time Monitoring

Overview Of Down Time Monitoring.

Down Time Monitoring is performed in Prodigy by a program known as Down Time Logger2. Down Time Logger is used to provide an accurate down time or utilisation logging facility for anything from test equipment through individual machines to whole production lines. This document will refer to equipment even though it may be one machine or a whole production line that is being monitored as the principal is the same. Whenever the equipment stops, Down Time Logger will start accruing down time and the reason for the stoppage can be allocated to that period of time. Down Time monitoring uses a suite of programs which will be described in this section .

Down Time Logger is configured using the Down Time Logger configuration program. This allows the user to indicate signals that will define when the equipment stops, whether it is in auto or manual mode and an interlock. The user also needs to define the possible down time reasons that can be assigned to stoppages.

By using an interlock mechanism the operator is obliged to input a reason for the down time as they are prevented from starting again until a valid down time reason has been entered. This means that the down time is allocated to a reason at the time that it happens, rather than at the end of a shift. With some products, however, many short periods of down time may be the norm and use of an interlock would actually slow production down. Therefore, Down Time Logger can be configured to record these short stoppages but not require reasons to be assigned to them and only when a significant stoppage is encountered will the interlock be set.

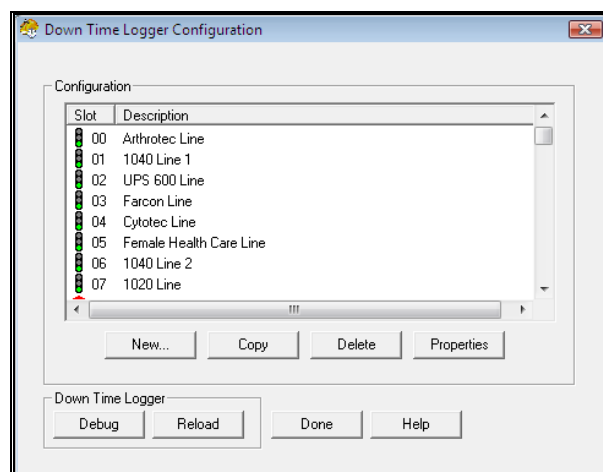
In some cases the time taken to enter the reason for the stoppage will have a significant impact on the production time and, therefore, a second interlock scenario can also be configured. Here a reason for the stoppage can be entered either during the stoppage itself or during the running period that immediately follows the stoppage. If a reason isn't entered and the equipment is stopped again the interlock will then be triggered, forcing reasons to be entered for both the previous and current stoppage.

Down Time Logger Configuration

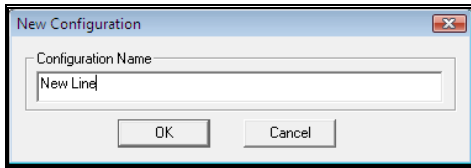
For each piece of equipment there should be a Down Time Configuration file. Each file is allocated a slot number and an icon similar to a traffic light shows whether it is currently active or not. A red light means it is not currently active so no Down Time Monitoring will be done on this configuration. A green light means it is active so when the Down Time Logger program is running this equipment will be monitored for Down Time as defined by its configuration parameters.

Configuration Selection

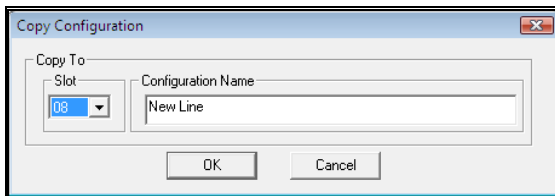
The Down Time Configuration files are stored in a standard area within the Prodigy directory structure. The Down Time Configuration program scans this area and presents a list of all of the configuration files present.



- **New.**
Select an empty slot and create a new configuration. A dialog will prompt for the name of the new configuration.

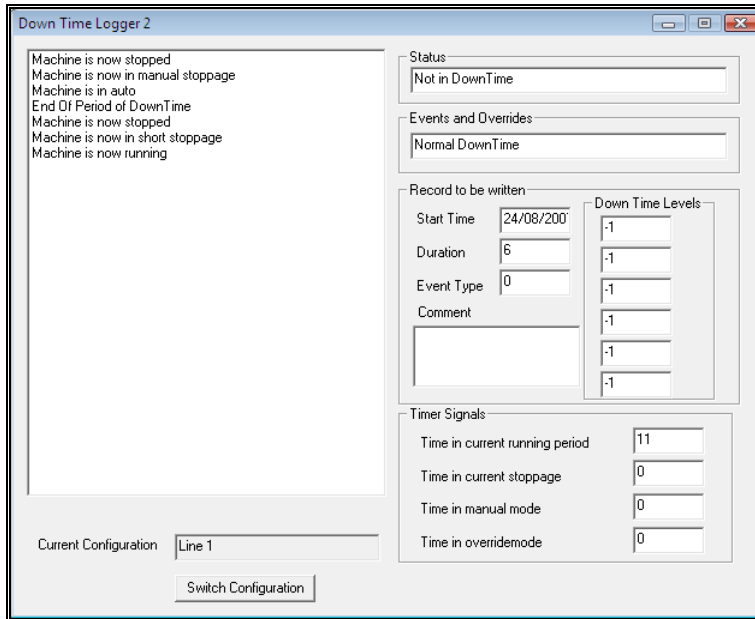


- **Copy.**
Create a copy of an existing configuration by selecting a configuration and pressing the **Copy** button. A dialog will prompt for a slot number and the name of the new configuration. The slot number must be one which is currently unallocated to a configuration. The existing configuration will be copied under the new name.

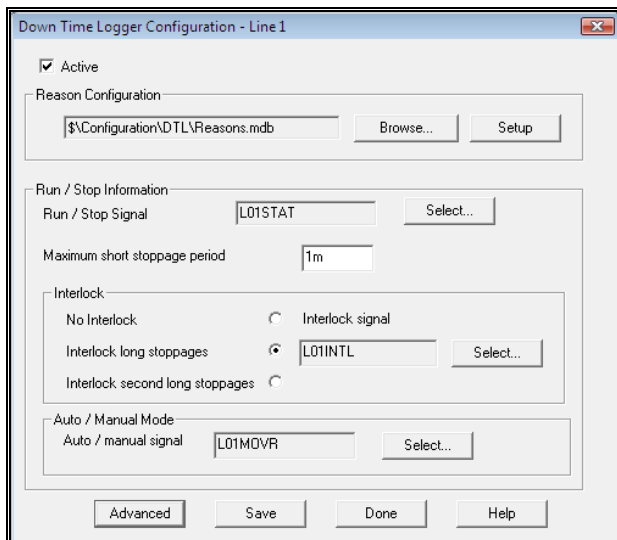


*When either a **New** or **Copied** configuration is created, the configuration file is written away with the **Active** flag set to false. This is so Down Time Logger will not try to process the configuration until the user has finished working on it (and ticks the **Active** flag).*

- **Delete.**
Existing configurations can be deleted by selecting them in the list and pressing the Delete button. A confirmation is requested before deleting the configuration.
- **Properties.**
By double clicking on the required selection (or by single clicking and then pressing the Properties button) the user is presented with a dialog that allows them to set or alter aspects of the currently selected configuration. See Configuring Properties below.
- **Reload.**
This option starts the Down Time Logger program if it is not already running, or stops and restarts it if it is already running. If a configuration is modified Down Time Logger must be restarted in order for the modifications to take effect. Note when Down Time Logger is restarted and the equipment is already stopped it will start again with a short stoppage. It will not remember the previous stoppage or the reason if any which was entered.
- **Debug.**
If this option is pressed and the Down Time Logger program is running it will display a Debug Window. This will show at a glance the Down Time Situation on any active configuration. Use the **Switch Configuration** button to show the information for another configuration. The data on the right relates to the currently selected configuration but the information on the left relates to all currently active configurations.



Configuring Properties



- Reason Configuration.**
 This specifies the name of the database in which the Down Time Reasons are held. One or many different Down Time configurations may use this database. You must configure one of these in order for Down Time to work. Press the browse button to search for this database. Pressing Setup invokes the Down Time Reason Configuration program, which is used to create and modify this database and this, is described later.
- Run Stop Signal.**
 This specifies the name of the digital signal that is used to inform Down Time Logger when the equipment is running or stopped. The equipment is stopped when the signal is 0 and running when the signal is 1.
- Maximum Short Stoppage Period.**
 Any stoppages less than this time are deemed to be insignificant and although they will be logged to the database and labelled short stoppage they do not need to have any down time reason associated with them. Stoppages greater than this time are labelled down time and do require a reason to be assigned to them.
- Interlock**
 There is a choice of three ways for the Down Time Logger to apply an interlock.

- **No Interlock.**

This means that when the equipment has stopped no Interlock will ever be applied. This means that once stopped the operator will not be forced to enter a reason.

- **Interlock Long Stoppages.**

Once the short stoppage period has elapsed if this option is used, the interlock is applied and the operator must enter a reason for the stoppage in order to restart the equipment.

- **Interlock Second Long Stoppages.**

This allows the operator more flexibility. If the equipment is stopped and the short stoppage period has elapsed the interlock will not be applied and the equipment may be restarted without the operator entering a reason for the stoppage. During the running phase the operator will still be able to assign a reason to the previous stoppage. If a reason hasn't been entered and the equipment stops for a second time the interlock will be applied immediately (it will not wait for the short stoppage period to elapse). The operator must now enter a Down Time Reason for both the previous stoppage and the current stoppage before the interlock will be cleared.

- **Interlock signal.**

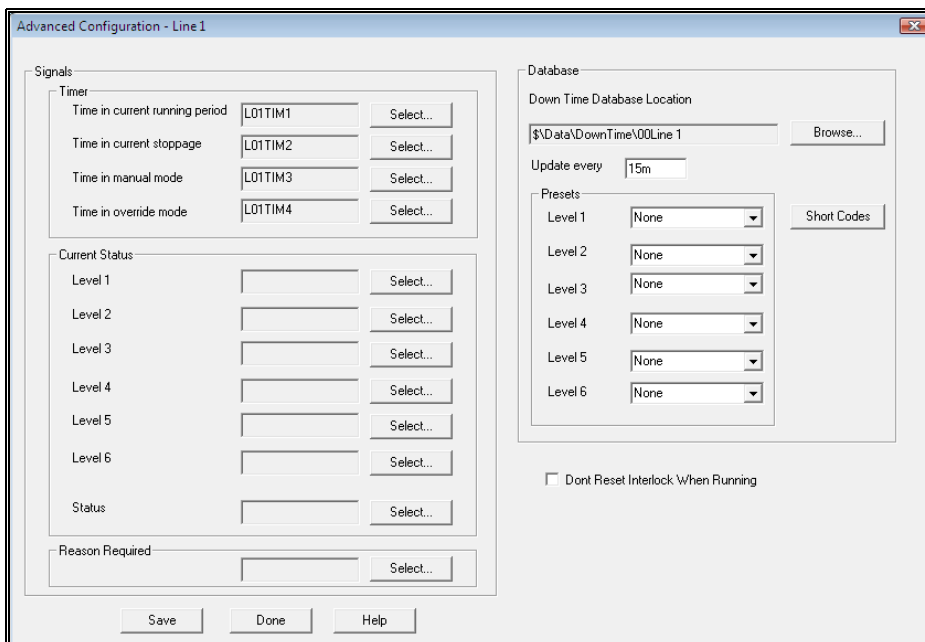
This specifies the name of the digital signal that is used by Down Time Logger to prevent the equipment from being restarted until a valid Down Time Reason has been entered. This signal must be entered if either Interlock Long Stoppages or Interlock Second Long Stoppages have been ticked. Down Time Logger sets the signal to 1 to set the interlock and 0 to clear the interlock.

- **Auto/Manual signal.**

This is optional and specifies the name of the signal that is used to inform Down Time Logger if the equipment is being used in a manual mode and, therefore, all down time should be allocated to a manual time category. Down Time Logger assumes that when this signal is 1 the equipment is in automatic mode and when it is 0 it is in manual mode. When in Manual mode the interlock is overridden.

Advanced Properties

Pressing the Advanced button allows the user more flexibility over configuring the Down Time System. Using these options is purely optional. The Down Time Logger may be used perfectly well without ever configuring any of these options.



- **Timer Signals.**

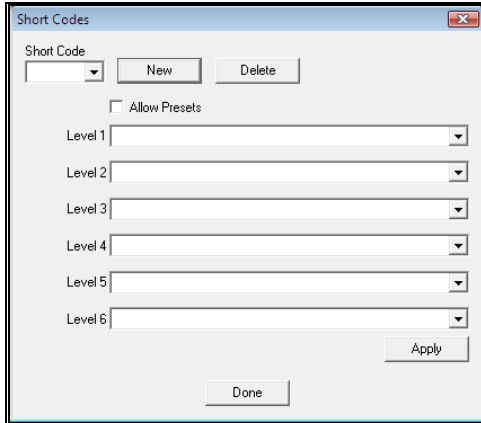
These signals can be configured to be continually updated with the time the equipment is in a particular state, either **Running**, **Stopped**, **Manual** and **Override**. These signals can then be put on a display to give the operator more information.

- **Current Status Signals.**
 These signals allow a mimic display to be updated with the current down time status for the equipment. The Level 1 to Level 6 signals show the currently entered down time reason as a set of Reason ID codes as set up in the Down Time Reason Configuration program. The Status signal should be either an event text or finite state signal and shows the current status for the equipment (Down Time Logger writes the textual state into slot 1 of this signal). The states displayed are :
 - No Reason Required
 - Override
 - Short Stoppage
 - Previous Reason Required
 - Current Reason Required
 - Current Reason Entered
- **Down Time Database Location.**
 This is the folder that the access databases are stored in that contains the Down Time Data. A Database is created when needed or at midnight. Thus if Down Time has occurred there will be a separate database for each day. Unless configured differently by the user there will be a separate folder for each Down Time configuration made up of the slot and name of the configuration. This option should only be used if the user wants to override this for a particular reason. For example to put all Down Time Data in a single database.
- **Update Every.**
 This determines the frequency with which the down time data is written. The value entered should be in the standard Prodigy time period format. In the example shown 15m would have the effect that every 15 minutes the down time data would be updated in the file. If the Down Time State changes such as a new reason being entered or going from a Short Stoppage to a significant Down Time or into Manual Mode the data is written at this time also.
- **Presets.**
 When the equipment first stops a record is written into the Down Time Database. Until the operator enters a reason for this stoppage, or if it is a short stoppage, there is no reason allocated to this Down Time record. The Presets allow the user to choose a default reason to be input. This will be written to the database until another reason overwrites it. Having a preset does not stop the operator from having to enter a current reason in the normal way or stop the interlock working. It is purely to put an initial Down Time Code in one or more levels.
- **Short Codes.**
 See below.
- **Don't Reset Interlock When Running.**
 A situation can occur where an interlock is set by the system just after a machine is started by an operator, the interlock is then active while a machine is running. This situation is undesirable as it is a state that the system should not be able to get into; but due to the difference in time between the start button being pressed and a machine starting and send out a running signal it is possible this may occur.

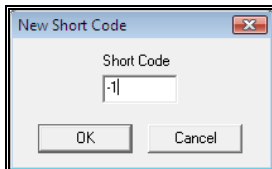
 Leaving this box unchecked causes the system to cancel an interlock that is has set if it receives a running signal from the machine after setting an interlock. If the box is checked the interlock will remain in place and depending on how the interlock relay is wired into the machine it may stop the machine or allow it to continue running.

Short Codes

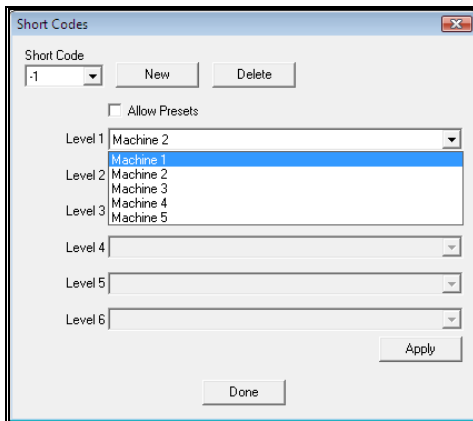
Short codes allow a quick way for the operator to enter a common downtime reason. Rather than have to peck through each level entering the relevant code, the operator can enter a single code that defines the reason for each of up to six levels. It therefore stands to reason that the short codes should be the most common down time events that occur on each line. The screen is shown below.



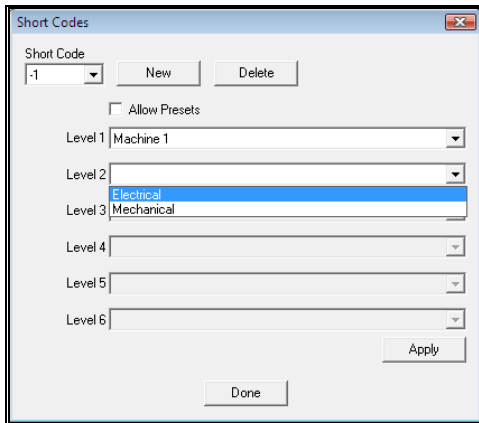
To create a new short code click the **New** button. A dialog is displayed that requests the identifier number for the new code. This should be a negative integer (-1, -2, -3 etc.).



After the code is entered you can select the down time reason for each of up to six levels. The dropdown lists display the event codes that are available for that level, and each successive level's codes depend on what has been selected further up in the hierarchy. An example of the level one list is shown below...



And the level two list is shown below...



After defining the reasons at the (up to) six levels, press the **Apply** button to save the changes.

To select an existing short code simply select its number from the dropdown list of Short Codes. The (up to) six levels of down time codes that have been defined for that short code will be displayed. You can alter the down time code for any of the existing levels, and doing so will cause the lists lower down the hierarchy to be invalidated and filled with a new set of possible selections. Once you have made all the changes you want, press **Apply** to save the changes.

To delete an existing short code, first select it as described above and then click the **Delete** button.

In operation, when a downtime reason now occurs the operator can simply enter -1 etc. at the down time terminal and this will allocate the correct downtime codes for the (up to) six levels defined.

Down Time Logger

The Down Time Logger program may be configured to start from the Optional Services button on the Prodigy Services screen. This documentation relates to Down Time Logger2.

It will read any active configurations that have been set up using the Down Time Configuration Program described above and monitor each piece of equipment as specified by each configuration. A piece of equipment will be deemed to be running if the Run/Stop signal is 1 and stopped if this signal is 0. When in the stopped state it is considered to be in Down Time. If any Timer signals have been defined for the running or stopped state they will update and reset appropriately.

The equipment may be in one of six states.

- **No Reason Required.**
The equipment is running and no down time reason is required for any previous stop.
- **Override Mode.**
The equipment may be running or stopped and the last reason entered was an override reason. The equipment will stay in Override Mode until Override is explicitly ended. If the Timer signal for Override has been defined it will update.
- **Short Stoppage.**
The equipment has been stopped for a short amount of time without a reason being entered and no reason is currently needed. One may be entered if required.
- **Previous Reason Required.**
The equipment is running but no reason was entered for the last stoppage of a significant time. Alternatively if the Interlock second long stoppages is configured the machine may be stopped with no reason having been entered for either the current stoppage or the previous one.
- **Current Reason Required.**
The equipment has been stopped for a significant amount of time but no reason has yet been entered.
- **Current Reason Entered.**
The equipment is stopped and a reason has been entered for the current stoppage. While the equipment is stopped an alternative reason may be entered if required. The duration of the first reason will be from the start of the stoppage to the entry of the second reason. The duration of the

second reason will be from the entry of the reason to either the time that the equipment is restarted or another reason is entered. There is no limit to the number of reasons that can be entered during the stoppage period.

Each portion of Down Time is logged in an Access database. There is one database per day when down time has occurred. If no down time has occurred for a particular day there will be no database for it. At midnight if a new database is required the old one is closed and a new one created. If it is not needed at midnight, as there is currently no down time it will be created when and if it is required.

Each portion of down time recorded has an event type associated with it. These are

- **Short Stoppage.**
The down time is less than the maximum short stoppage time and does not need to have a reason associated with it.
- **Down Time.**
This is stoppage which is greater than the maximum short stoppage time. It should have a down time reason allocated to it when the operator has entered it.
- **Override.**
Overrides are used to disable the interlock for a period of time, for example when changing from one product to another or setting up the equipment. Once in an override mode the time is accumulated whether the equipment is running or stopped until the override is explicitly ended.
- **Manual.**
Any stoppages while in Manual Mode are logged and a down time reason should be entered for the stoppage. Whilst in manual mode the interlock will not be applied.

Typical Down Time Scenarios

The following are typical down time scenarios, which will show how the Down Time Logger deals with different types of down time and different ways of configuring the Down Time Logger.

Typically the equipment is running (As defined by the Run/Stop Signal). *No Reason required.* It then stops and goes into *Short Stoppage.* If it starts to run before the Maximum short stoppage period is reached it will go back into *No Reason required.* This stoppage will be logged against *Short Stoppages.* If however it is still stopped when the Maximum short stoppage period is reached it will go from a short stoppage period of time to *Current Reason required.* The whole of this stoppage will now be logged against the event *Down Time.* If the interlock long stoppages has been selected the interlock will now be applied preventing the equipment from running until a reason for this down time has been entered. Once a reason has been entered the state will change to *Current Reason Entered* and once the equipment has started running back to *No Reason required.*

If at the point the stoppage starts to be logged against *Down Time* the interlock second long stoppages has been selected the interlock is not yet applied and the operator is free to start running. The state changes to *Previous Reason Required.* If however the equipment stops again before a reason has been entered for this previous stoppage the interlock will be applied immediately. This stoppage will be logged against a new *Down Time* record. In order to release the interlock and start the equipment the operator must enter two down time reasons, which may be the same, one for the previous stoppage (the status will change to *Current Reason required*) and one for this stoppage. Once the second reason has been entered the status will change to *No Reason required* and the interlock will be released.

Although a short stoppage doesn't need a reason the operator may put one in. If they do the stoppage will be logged as *Down Time.*

If at any point the Auto/Manual signal (If it has been defined) goes to 0 signifying manual. It allows the interlocks to be overridden. Any Down Time will now be logged against *Manual* in the database. If the Timer signal for manual time has been defined it will start clocking up the time the equipment is in manual mode. Manual time will only be logged as down time if the equipment is stopped.

When a reason is entered and this reason has been configured as an override reason the down time will be logged against this override reason. It will have the effect of releasing the interlock. Time will start to accumulate regardless of whether the equipment is running or stopped until the override is ended by the operator.

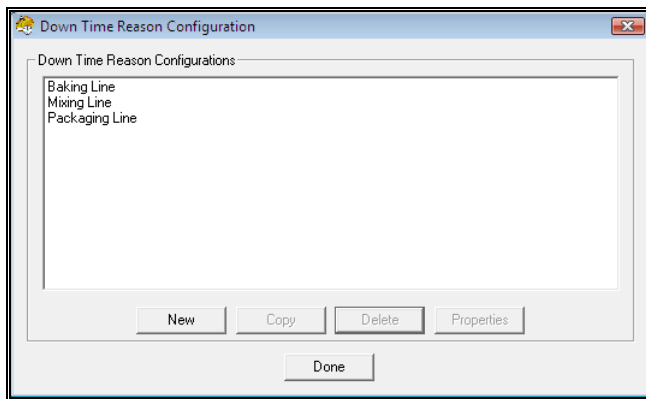
Down Time Reason Configuration

Down Time Reasons specify, as the name would suggest, the reasons for a particular down time period that occurs. They form a hierarchical structure that branches further and further out the more levels you go down.

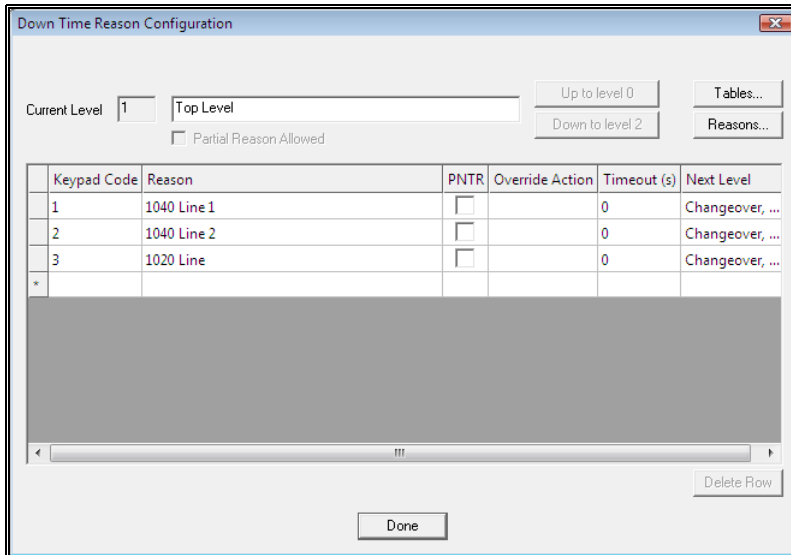
- In a simple system there may, for example, be two top-level down time reasons, Planned and Unplanned. Below Planned there may be further reasons such as Maintenance, Factory Shutdown, Changeover and so on. Below Unplanned there may be things like Electrical Failure, Belt Jam and so on. Choosing the correct down time reason structure is crucial, ensuring that you go down to sufficient detail to cover all the eventualities that might occur whilst attempting to keep the structure from becoming too large or unwieldy.
- If you have multiple lines and wish to produce reports that cover all of the lines it is beneficial to define one set of downtime structures that cover all the lines. In this instance you could decide to make the top level down time reasons be the line name, e.g. Mixing Line, Baking Line, Packaging Line. Alternatively if you are only ever going to produce reports on individual lines they can use separate Down Time Reason configurations.
- When entering down time reasons via either a shop floor terminal or a PC the operator specifies the relevant down time reason at each level. Our example would be “Mixing Line, Unplanned, Belt Jam”. Levels can be skipped with preset entries. For example, if a shop floor terminal is being used for multiple lines, each function key could represent a different line; Function key 1 would be for the Mixing Line, function key 2 would be for the Baking Line etc.
- When generating reports of down time the amount of down time that occurred can be accumulated to any level. If you report to level 3 (in our simple example above) you would see how much down time was attributed to Belt Jam, Electrical Failure etc. If you reported to level 2 you would see how much down time is attributed to Planned and Unplanned.

Bear all of these points in mind when configuring down time reasons.

Down Time Reason Configuration Program



The main screen displays a list of Down Time Reason configurations. You can create, copy and delete configurations by clicking the relevant buttons and you can define the properties of a configuration by clicking the **Properties** button or double-clicking on the required configuration in the list. The Reasons configuration is shown below.



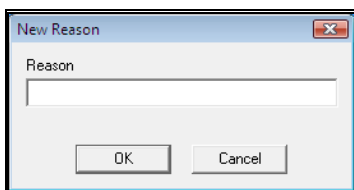
As was mentioned elsewhere, the down time reasons configuration is a hierarchical tree structure so for each row in the table that is currently being viewed there is the option to access or create a sub-table. The current level of table that you are on is shown in the top left of this screen (**Current Level**) and alongside that is the name of the current table. The name can be modified by altering the text within this box, except for the top level (level 1) which is always called **Top Level**.

You create extra rows in the table by clicking on the * field at the left of the bottom most row. You then enter whatever data is required into the row. The data is as follows:

- Operator Code.**
 This is a numeric code that is used to select the option specified by this row when it appears on a shop floor terminal. All of the reasons shown in this dialog will be displayed in an order that is dictated by the operator code (numerically smallest to largest). It makes sense to number the codes 1, 2, 3 etc. but it is equally valid to number them 10, 15, 20, etc. These codes are not used anywhere else in the system.
- Reason.**
 This is the reason to which down time will be attributed. Reason names must be unique at a particular level, though you can use an existing name at a different level. In the background this reason name is assigned a code that is unique throughout the system and it is this code that is used internally within the Down Time Monitoring system. The reason name is only looked-up from the code as and when required. This means that the name of the reason can be changed later (see elsewhere) but will still be associated with any data that has already been collected for that reason.

Keypad Code	Reason	PNTR	Override Action	Timeout (s)	Next Level
1	1040 Line 1	<input type="checkbox"/>		0	Changeover, ...
2	1040 Line 2	<input type="checkbox"/>		0	Changeover, ...
3	1020 Line	<input type="checkbox"/>		0	Changeover, ...
*	New...				
	1020 Line				
	1040 Line 1				
	1040 Line 2				

You select existing reasons from the drop-down list or select the **New...** option (on the drop-down list) to create a new one. This will display a dialog that asks for the name of the new reason, which must be unique within the current table level.



- **Override Action.**
Select the required override action (if any) from the drop-down list

Keypad Code	Reason	PNTR	Override Action	Timeout (s)	Next Level
1	1040 Line 1	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
2	1040 Line 2	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
3	1020 Line	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
*			None Override		

- **Next Level Table.**

Keypad Code	Reason	PNTR	Override Action	Timeout (s)	Next Level
1	1040 Line 1	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
2	1040 Line 2	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
3	1020 Line	<input type="checkbox"/>		0	Changeover, Planned or Unplanned
*					New... Changeover, Planned or Unplanned

Select the next level table name from the drop-down list or select New... (from the drop-down list) to create a new table. Selecting New displays a dialog requesting a name for the new table, the name must be unique at the level at which it will be created (the level below the current one). As with reasons, the table name may be altered later and anything that accessed that table will show the new name.

- **Tables.**
This button displays a dialog that contains all of the tables that have been defined at the level below the current one (i.e. all tables that can be linked to from this level).

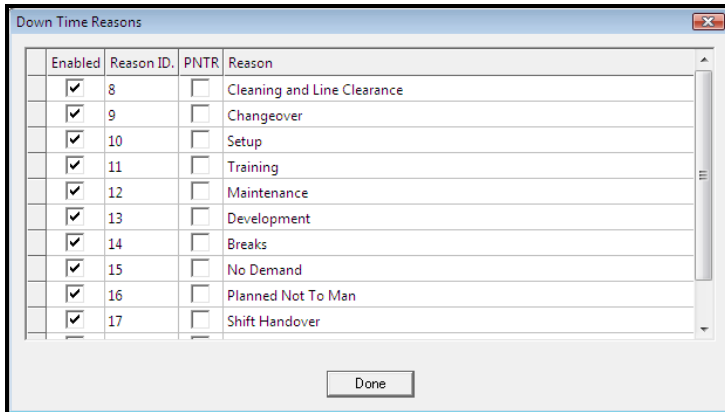
Enabled	Table No.	Level	Table Name	Allow Partial
<input checked="" type="checkbox"/>	2	3	Changeover	<input type="checkbox"/>
<input checked="" type="checkbox"/>	3	3	Planned Down Time	<input type="checkbox"/>
<input checked="" type="checkbox"/>	4	3	Unplanned Down Time	<input type="checkbox"/>
<input checked="" type="checkbox"/>	5	3	Unplanned Stops	<input type="checkbox"/>

Done

- **Enabled.**
Unchecking this box means that the table will no longer be available in the drop-down list of *Next Level Tables*.
- **Table No & Level.**
These fields cannot be modified and are shown for reference only.
- **Table Name.**
This is the name of the table and may be modified at will simply by typing a new name into the relevant field (max 32 characters). Changing the name will not affect any linkage that may have already been defined, but anywhere that used the old table name will use the new one instead.
- **Allow Partial.**
Checking this box means that when the operator is specifying the down time reason for the current stoppage via the shop floor terminals they can stop at the current level. If the box is not checked they must proceed to the next level and so on, until they either reaches a level that has Allow Partial checked or is a bottom level. It is up to the operator (and circumstances) as to whether they stop at the point where partial entry is allowed or proceeds to fully specify the reason.

- **Reasons.**

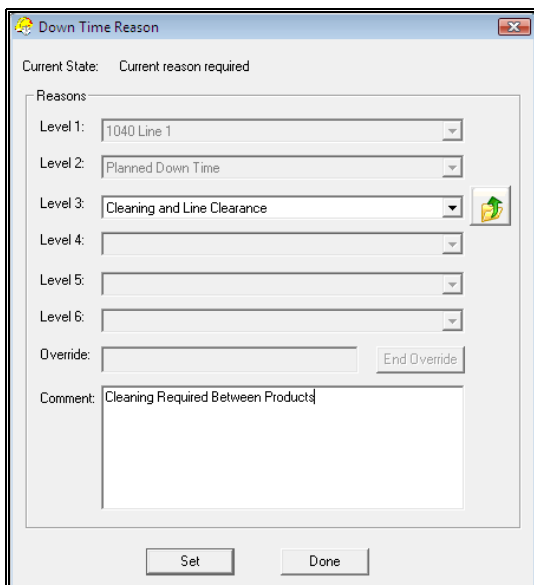
This is rather similar to the Tables function described above. A dialog is shown that contains all of the reasons that are valid for the current table level.



- Again, unchecking the **Enabled** box means that a reason is no longer shown on the **Reasons** drop-down list
- The reason name can be changed by typing a new name into the relevant field on the dialog. Anywhere that the old name had been used will subsequently show the new name.
- **Reason ID** is the code that is logged into the Access database against the relevant level and is shown if the Down Time Logger's Current Status signals are being used.
- **Level** is shown only for reference.
- **Down To Level n.**
The Up and Down level buttons traverse the table configuration tree by displaying and allowing the configuration of one node of the tree at a time. The Down To Level n button is only available when you have highlighted a row on the reasons list and when you click the button the configuration for the next level table (as specified in the right-hand most column Next Level Table) is displayed.
- **Up To Level n.**
This button displays the configuration for the table above the current level and is available on all levels except the top level.

Down Time Logger Reason

DTL Reason allows a user to view and set down time reasons similar to the way that IDP terminals can.



The current state of down time for the chosen slot is displayed at the top of the dialog. There then follows the down time reasons that can be chosen. As each level is chosen, the reason for the next level becomes available. When the down time reason tree has reached its end, or a partial reason is chosen, the Set button becomes available, and a comment can be entered. If the incorrect reason is chosen and you want to back up a level, there is an “up” button to the right of the down time reason. Pressing it will remove the reason at the current level and reverse up one level.

Press the *Set* button to set the reason..

If an *Override* reason is chosen the *End Override* button becomes available. Click this button to end the override state.

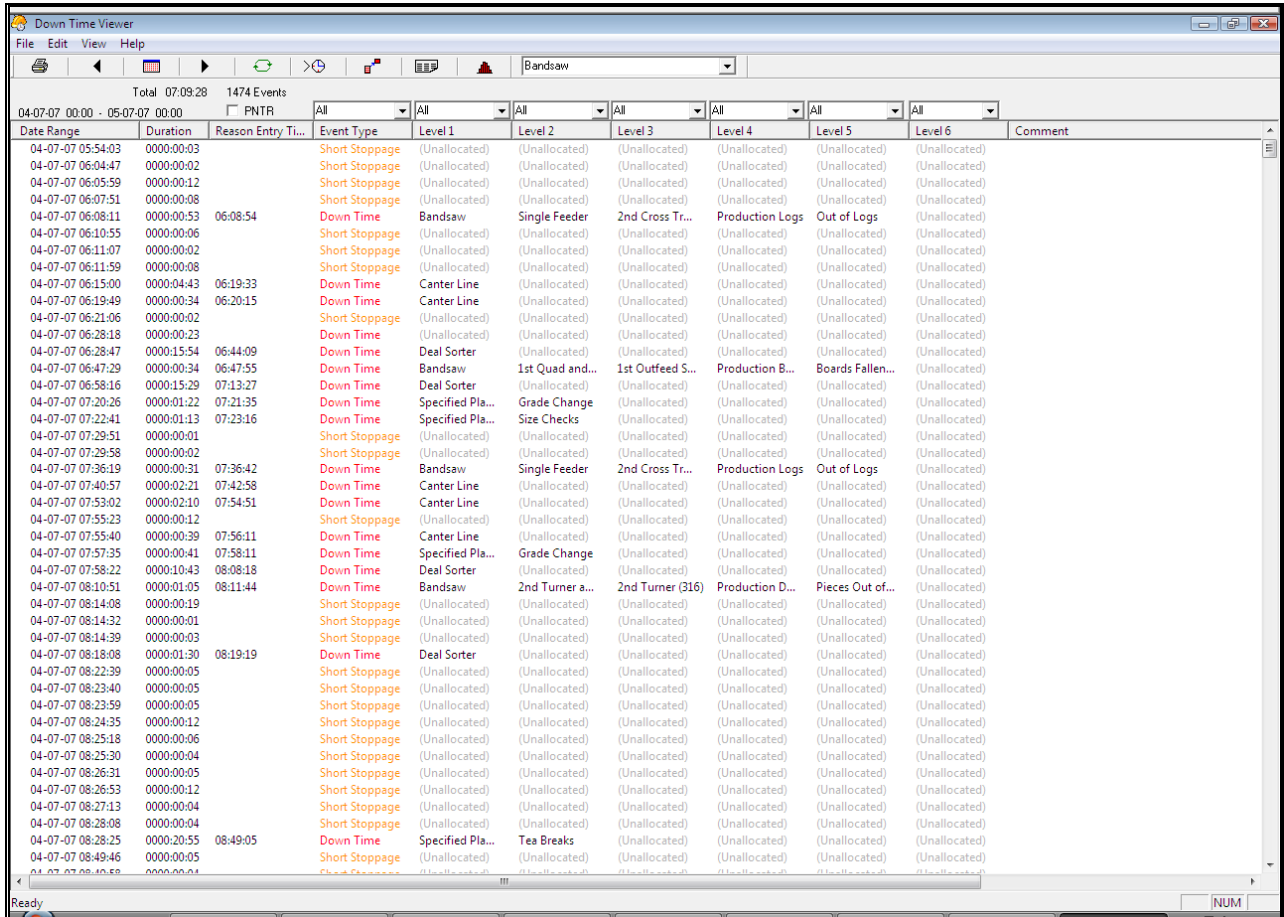
Click the *Done* button to dismiss the screen.

Switches that can be used when invoking DTL Reason

- **-view** – puts the software in view only mode.
- **-slot=<slot number>** - chooses the Down Time Reason slot to use the configuration of.
- **-preset1=<preset number>**
- **-preset2=<preset number>**
- **-preset3=<preset number>**
- **-preset4=<preset number>**
- **-preset5=<preset number>**
- **-preset6=<preset number>** - choose the presets for DTL levels. This “pre-chooses” reasons at the specified levels.

Down Time Viewer

The Down Time Viewer allows you to interactively view and modify down time data. Data is viewed by specified chunks of time and can be filtered by a variety of methods. Data manipulation includes the ability to combine and split down time records and to assign and/or change downtime reasons for existing records. In addition there are some simple reporting options available as well as the facility to export the data to CSV file for viewing in Excel™. The Down Time Viewer screen is shown below and the use of the program is described in full in the following section.



The screen has a toolbar along the top that controls how the data is viewed. Below that is an information area and a second toolbar that controls which down time events are displayed. Below that is the main display area of the program in which all of the selected down time reasons are displayed.

The Display Area Explained

The display consists of a number of columns and within these columns a variety of symbols and colours are used to indicate certain conditions. Moving from the left most column to the right we see:

Date Range

The **Date Range** column can contain a number of coloured icons that indicate that some records display a contiguous period of downtime. This is best illustrated by example. If a downtime period lasts for 5 minutes, 1 minute of it may be attributed to electrical fault, followed by 2 minutes of mechanical failure followed by 2 minutes of warm-up time. These three reasons were all part of the same downtime event, and so are marked as being contiguous. Icons in the Date Range column show which records are contiguous with which others. These icons are as follows

- Contiguous with a record that appears above this one, but off the screen.

Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2
↑ 04-07-07 16:49:47	0000:00:22	16:49:47	Down Time	Specified Pla...	Size Checks
04-07-07 16:57:03	0000:00:10		Short Stoppage	(Unallocated)	(Unallocated)

- Contiguous with the record above (or below).
These always appear as pairs (or groups, see below) as you cannot have one record be contiguous with the one above without the one above being contiguous with the one below itself, if you follow.

04-07-07 19:59:32	0000:00:22		Down Time	(Unallocated)	
↓ 04-07-07 19:59:57	0000:00:05	20:00:00	Down Time	Edger 1	
↑ 04-07-07 20:00:01	0000:00:09	20:00:01	Down Time	Edger 2	
04-07-07 20:00:16	0000:00:02		Short Stoppage	(Unallocated)	

- Contiguous block.
This is as for the situation described above, but there are three or more records forming part of the contiguous block. The block will begin with a down arrow and end with an up arrow and between these can be any number of up-and-down arrows.

04-07-07 19:13:46	0000:00:44	19:14:17	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)
↓ 04-07-07 19:14:33	0000:21:57	19:34:49	Down Time	Specified Pla...	Tea Breaks	(Unallocated)	(Unallocated)	(Unallocated)
↑ 04-07-07 19:36:29	0000:01:39	19:36:29	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs
↓ 04-07-07 19:38:07	0000:00:09	19:38:07	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 19:38:19	0000:00:42	19:38:47	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs

- Contiguous with the record that appears below this one, but off the screen.

04-07-07 19:13:46	0000:00:44	19:14:17	Down Time	Specified Pla...	Size Checks
↓ 04-07-07 19:14:33	0000:21:57	19:34:49	Down Time	Specified Pla...	Tea Breaks

Duration

The **Duration** column shows how long the down time event lasted, and on each screen there will be one duration that is marked in a different colour. This duration is the longest duration on the current screen.

The **Reason Entry Time** column shows the time at which the reason was entered. Normally this is just displayed as a time, but it is possible that it will show a date and time (see below).

04-07-07 08:28:08	0000:00:04		Short Stoppage	(Unallocated)	(Unallocated)
↓ 04-07-07 08:28:25	0000:14:55	08:49:05	Down Time	Specified Pla...	Tea Breaks
↑ 04-07-07 08:43:19	0000:06:00	12-12-07 14:51:45	Down Time	Specified Pla...	Saw Changes
04-07-07 08:49:46	0000:00:05		Short Stoppage	(Unallocated)	(Unallocated)

This means that the reason was entered on a different date to which the downtime event occurred, and in some cases could be several days or weeks later, if the record is entered or modified after the event by using Down Time Viewer.

Event Type

The **Event Type** column shows the type of event (*manual*, *short stoppage*, *down time* or *override*) and each type is displayed in a different colour.

04-07-07 06:11:59	0000:00:08		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:15:00	0000:04:43	06:19:33	Override	Center Line	(Unallocated)
04-07-07 06:19:49	0000:00:34	06:20:15	Down Time	Center Line	(Unallocated)
04-07-07 06:21:06	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:28:18	0000:00:23		Down Time	(Unallocated)	(Unallocated)
04-07-07 06:28:47	0000:15:54	06:44:09	Manual	Deal Sorter	(Unallocated)
04-07-07 06:47:29	0000:00:34	06:47:55	Down Time	Bandsaw	1st Quad and...
04-07-07 06:58:16	0000:15:29	07:13:27	Manual	Deal Sorter	(Unallocated)
04-07-07 07:20:26	0000:01:22	07:21:35	Down Time	Specified Pla...	Grade Change
04-07-07 07:22:41	0000:01:13	07:23:16	Down Time	Specified Pla...	Size Checks
04-07-07 07:29:51	0000:00:01		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 07:29:58	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 07:36:19	0000:00:31	07:36:42	Down Time	Bandsaw	Single Feeder

Override events are black, Manual events are blue, Down Time events are red, and Short Stoppages are orange.

Levels

The columns **Level 1** to **Level 6** show the down time reason entered for each of these levels. Where the down time is unallocated, the field is shown greyed-out so that fields where there are data stand out more.

Comment

The **Comment** column shows the comment (if any) that was entered along with the down time reason.

Filtering

You can filter the data in a variety of ways.

Filtering by Configuration

The first of these is by configuration. A drop down list contains the names of all Down Time Logger configurations that have been created (this list is the item with **Bandsaw** in it, below).



From the name of the configuration, Down Time Viewer finds the database into which down time records have been recorded. This quite possibly (though it is not mandatory) is a different database for each configuration. Thus, selecting the Down Time Configuration performs the first level of filtering.

Filtering by Event & Levels

Next you can filter by Event Type and/or Levels by selecting the relevant items from the row of dropdown lists just above the display area (below).

Total 07:09:28 1474 Events									
Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
04-07-07 05:54:03	0000:00:03		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:04:47	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:05:59	0000:00:12		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:07:51	0000:00:08		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:08:11	0000:00:53	06:08:54	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)
04-07-07 06:10:55	0000:00:06		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:11:07	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:11:59	0000:00:08		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:15:00	0000:04:43	06:19:33	Down Time	Center Line	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:19:49	0000:00:34	06:20:15	Down Time	Center Line	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:21:06	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:28:18	0000:00:23		Down Time	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:28:47	0000:15:54	06:44:09	Down Time	Deal Sorter	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 06:47:29	0000:00:34	06:47:55	Down Time	Bandsaw	1st Quad and...	1st Outfeed S...	Production B...	Boards Fallen...	(Unallocated)

The image below shows the list filtered on Event Type “Down Time”.

Total 06:16:19 126 Events									
Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
04-07-07 17:26:36	0000:00:19	17:26:44	Down Time	Specified Pla...	Grade Change	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 17:27:59	0000:01:08	17:29:02	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 17:34:28	0000:01:17	17:35:39	Down Time	Bandsaw	2nd Quad an...	Outfeed Side ...	Production D...	Deal Fallen O...	(Unallocated)
04-07-07 17:54:57	0000:01:01	17:55:48	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)
04-07-07 17:57:48	0000:02:28	17:59:15	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)
04-07-07 18:01:44	0000:07:40	18:09:02	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)
04-07-07 18:19:52	0000:00:19	18:19:59	Down Time	Specified Pla...	Grade Change	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:20:34	0000:02:01	18:21:36	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:25:08	0000:02:16	18:27:11	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)
04-07-07 18:33:22	0000:01:12	18:34:25	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:38:17	0000:00:37	18:38:41	Down Time	Specified Pla...	Grade Change	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:39:51	0000:01:14	18:40:57	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:48:39	0000:01:23	18:49:42	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)
04-07-07 18:51:21	0000:00:28	18:51:42	Down Time	Specified Pla...	Size Checks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)

The image below shows the list filtered on Event Type “Down Time” and Level 1 reason “Bandsaw”.

Total 01:58:38 45 Events										
04-07-07 00:00 - 05-07-07 00:00			PNTR	Down Time	Bandsaw	All	All	All	All	All
Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	
04-07-07 07:36:19	0000:00:31	07:36:42	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 08:10:51	0000:01:05	08:11:44	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)	
04-07-07 09:01:49	0000:02:53	09:02:26	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)	
04-07-07 09:39:31	0000:01:29	09:40:31	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)	
04-07-07 09:47:29	0000:02:03	09:48:05	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)	
04-07-07 10:12:19	0000:02:06	10:12:47	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)	
04-07-07 10:15:09	0000:07:20	10:16:15	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:28:25	0000:01:17	10:28:56	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)	
04-07-07 10:31:32	0000:00:20	10:31:55	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:32:00	0000:00:13	10:32:00	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:33:50	0000:01:09	10:34:49	Down Time	Bandsaw	1st Turner an...	1st Turner (308)	Production Logs	Log Jammed	(Unallocated)	
04-07-07 10:42:13	0000:00:51	10:42:53	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:47:34	0000:01:59	10:49:22	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 12:20:33	0000:01:55	12:22:10	Down Time	Bandsaw	2nd Turner a...	2nd Turner (316)	Production D...	Pieces Out of...	(Unallocated)	

The image below shows the list filtered on Event Type “Down Time”, Level 1 reason “Bandsaw” and Level 5 reason “Out of Logs”. Any combination of Event Type and Levels can be selected to arrive at the filtering that you require.

Total 00:15:06 10 Events										
04-07-07 00:00 - 05-07-07 00:00			PNTR	Down Time	Bandsaw	All	All	All	Out of Logs	All
Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	
04-07-07 06:08:11	0000:00:53	06:08:54	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 07:36:19	0000:00:31	07:36:42	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:15:09	0000:07:20	10:16:15	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:31:32	0000:00:20	10:31:55	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:32:00	0000:00:13	10:32:00	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:42:13	0000:00:51	10:42:53	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 10:47:34	0000:01:59	10:49:22	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 19:36:29	0000:01:39	19:36:29	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 19:38:19	0000:00:42	19:38:47	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	
04-07-07 21:50:03	0000:00:40	21:50:37	Down Time	Bandsaw	Single Feeder	2nd Cross Tr...	Production Logs	Out of Logs	(Unallocated)	

Filtering by Duration

You can filter by down time duration. Click the **Filter by Duration** button  to display the dialog shown below.

Select Reason Duration

Show reasons with duration greater than :

OK Cancel


Enter the minimum duration length that you want to filter by (the dialog is of the form hh:mm:ss) and click OK. The display will then show only those records that have a duration greater than the value entered here.

Durations > 00:20:00 Total 02:33:47 6 Events										
04-07-07 00:00 - 05-07-07 00:00			PNTR	All	All	All	All	All	All	All
Date Range	Duration	Reason Entry Ti...	Event Type	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	
04-07-07 08:28:25	0000:20:55	08:49:05	Down Time	Specified Pla...	Tea Breaks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	
04-07-07 11:01:00	0000:23:51	11:24:34	Down Time	Specified Pla...	Tea Breaks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	
04-07-07 11:27:11	0000:20:45	11:46:38	Down Time	Center Line	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	
04-07-07 13:19:05	0000:44:55	14:03:36	Down Time	Bandsaw	Deal Line Kic...	Kick Off Chai...	Production Misc	Line Full	(Unallocated)	
04-07-07 16:28:22	0000:21:26	16:49:27	Down Time	Specified Pla...	Tea Breaks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	
04-07-07 19:14:33	0000:21:57	19:34:49	Down Time	Specified Pla...	Tea Breaks	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	


Filtering by Selection

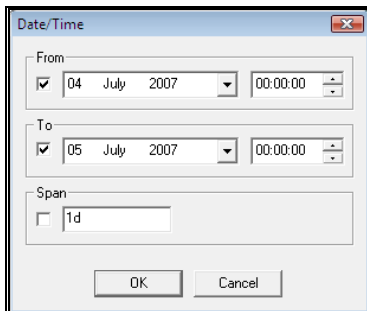
The final method of filtering is by selection. Select the individual records that you want to look at by clicking, shift+clicking and/or ctrl+clicking on the required records.

04-07-07 05:54:03	0000:00:03		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:04:47	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:05:59	0000:00:12		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:07:51	0000:00:08		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:08:11	0000:00:53	06:08:54	Down Time	Bandsaw	Single Feeder
04-07-07 06:10:55	0000:00:06		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:11:07	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:11:59	0000:00:08		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:15:00	0000:04:43	06:19:33	Override	Canter Line	(Unallocated)
04-07-07 06:19:49	0000:00:34	06:20:15	Down Time	Canter Line	(Unallocated)
04-07-07 06:21:06	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 06:28:18	0000:00:23		Down Time	(Unallocated)	(Unallocated)
04-07-07 06:28:47	0000:15:54	06:44:09	Manual	Deal Sorter	(Unallocated)
04-07-07 06:47:29	0000:00:34	06:47:55	Down Time	Bandsaw	1st Quad and...
04-07-07 06:58:16	0000:15:29	07:13:27	Manual	Deal Sorter	(Unallocated)
04-07-07 07:20:26	0000:01:22	07:21:35	Down Time	Specified Pla...	Grade Change
04-07-07 07:22:41	0000:01:13	07:23:16	Down Time	Specified Pla...	Size Checks
04-07-07 07:29:51	0000:00:01		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 07:29:58	0000:00:02		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 07:36:19	0000:00:31	07:36:42	Down Time	Bandsaw	Single Feeder
04-07-07 07:40:57	0000:02:21	07:42:58	Down Time	Canter Line	(Unallocated)
04-07-07 07:53:02	0000:02:10	07:54:51	Down Time	Canter Line	(Unallocated)
04-07-07 07:55:23	0000:00:12		Short Stoppage	(Unallocated)	(Unallocated)
04-07-07 07:55:40	0000:00:39	07:56:11	Down Time	Canter Line	(Unallocated)
04-07-07 07:57:35	0000:00:41	07:58:11	Down Time	Specified Pla...	Grade Change
04-07-07 07:58:22	0000:10:43	08:08:18	Down Time	Deal Sorter	(Unallocated)
04-07-07 08:10:51	0000:01:05	08:11:44	Down Time	Bandsaw	2nd Turner a...

Then pressing the **Filter by Selection** button . The display will change to show only those records that you selected.

Selecting A Date Range



By default the display shows 1 day's worth of downtime, a 24 hour period running from midnight to midnight. You can change this period by clicking the **Select Day**  button, which will display the dialog below.



The dialog box titled "Date/Time" has a close button in the top right corner. It contains three sections: "From" with a date selector (04 July 2007) and a time selector (00:00:00); "To" with a date selector (05 July 2007) and a time selector (00:00:00); and "Span" with a checkbox and a text field containing "1d". At the bottom are "OK" and "Cancel" buttons.

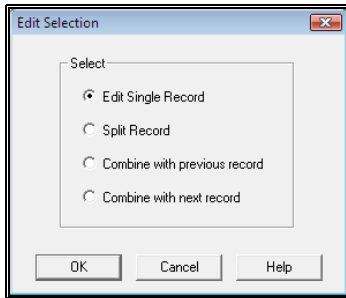
This allows you to specify the period by entering the **From** and **To** dates or by entering either the **From** or **To** dates and the **Span**. This not only sets the viewing span, but can also get you directly to the date(s) that you are interested in.

Stepping Through The Data

You can step backwards and forwards through the data by using the  and  buttons. This moves back or forward by whatever the currently selected time period is (usually 24 hours).

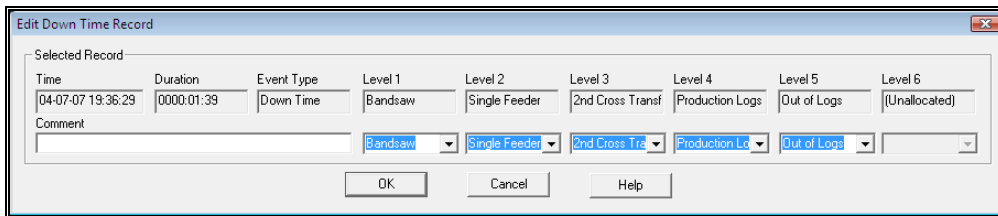
Editing Data

To edit a record double⁺click on the required row in the main screen, which will display the following dialog.



From this select the editing action that you want to perform.

Edit Single Record



This displays the current state of the record and below that gives fields allowing you to enter/change the down time reasons for each of the six levels. You can also enter a comment. On completion (when you click OK) the record will be rewritten with the new data and the Reason Entry Time will be set to the current date and time (visible on the screen, second line down in the image below).

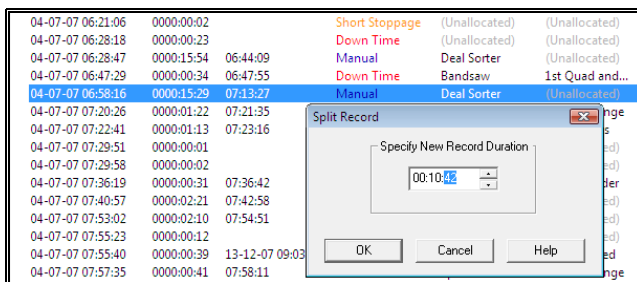
04-07-07 07:55:23	0000:00:12		Short Stoppage	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	
04-07-07 07:55:40	0000:00:39	13-12-07 09:03:33	Down Time	Specified Pla...	Not Manned	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	Modified for manual
04-07-07 07:57:35	0000:00:41	07:58:11	Down Time	Specified Pla...	Grade Change	(Unallocated)	(Unallocated)	(Unallocated)	(Unallocated)	

Combine With Previous/Next Record

The current record is combined with the next (or previous) record. This option is only available for events of the same type and if the records concerned are *contiguous* (and are therefore marked with blue arrows). The resulting record will have a duration equal to the duration of the two records that were combined. After combining you may need to perform an **Edit Single Record** to give the resulting record the correct down time reasons.

Split Record

This splits the current record into two. A dialog asks how much of the duration time should be given to the new record.

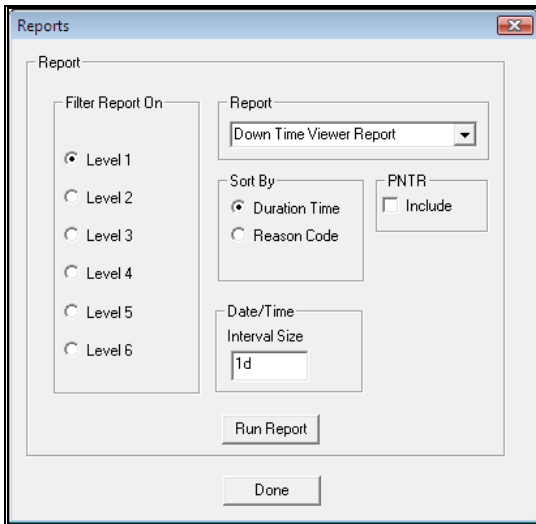


After splitting there will be two records whose total duration time is equal to the duration time of the original record. Naturally the two records will be contiguous (below).

↓ 04-07-07 06:58:16	0000:04:47	07:13:27	Manual	Deal Sorter	(Unallocated)
↑ 04-07-07 07:03:02	0000:10:42	07:13:27	Manual	Deal Sorter	(Unallocated)

Reporting & Exporting Data

Reporting and exporting is accessed via the Report  button. Clicking this button displays the dialog below.



Reporting

There is a drop down list of **Report Types** to choose from. These are simple, pre-configured downtime reports that will meet some of your basic needs. To do more in-depth reporting you will need to get into the realms of **Down Time Data Processing**, described elsewhere in this manual. The reports can be modified, and new reports written, so long as they only use the data that is generated by this reporting function (details given elsewhere). Any new reports that are written will automatically appear in the drop down list.

Select the report that you require from the drop down list and then select the level that you want to **Filter Report On**. If you select Level 1 the data is summed up into all Level 1 events; selecting Level 2 means the data is summed into all Level 2 events and so on.

Select how you want the data to be sorted by clicking the relevant button in the **Sort By** section. If you select **Greatest Time** then the data will be displayed with the down time event that had the greatest amount of time allocated to it in the left-most position, followed by the next highest amount and so on. This is useful because due to space restrictions not all down time reasons can be shown. Selecting this option ensures that the ones with the highest amount of down time are the ones that are seen.

Sorting by Reason Code ensures that the data is always output in the same order (i.e. the same reason code will always be first) no matter how much time is allocated to that reason. This is not true of the other sorting methods, and means that you can use this sorting method for reports that state how much time was allocated to specific down time reasons.

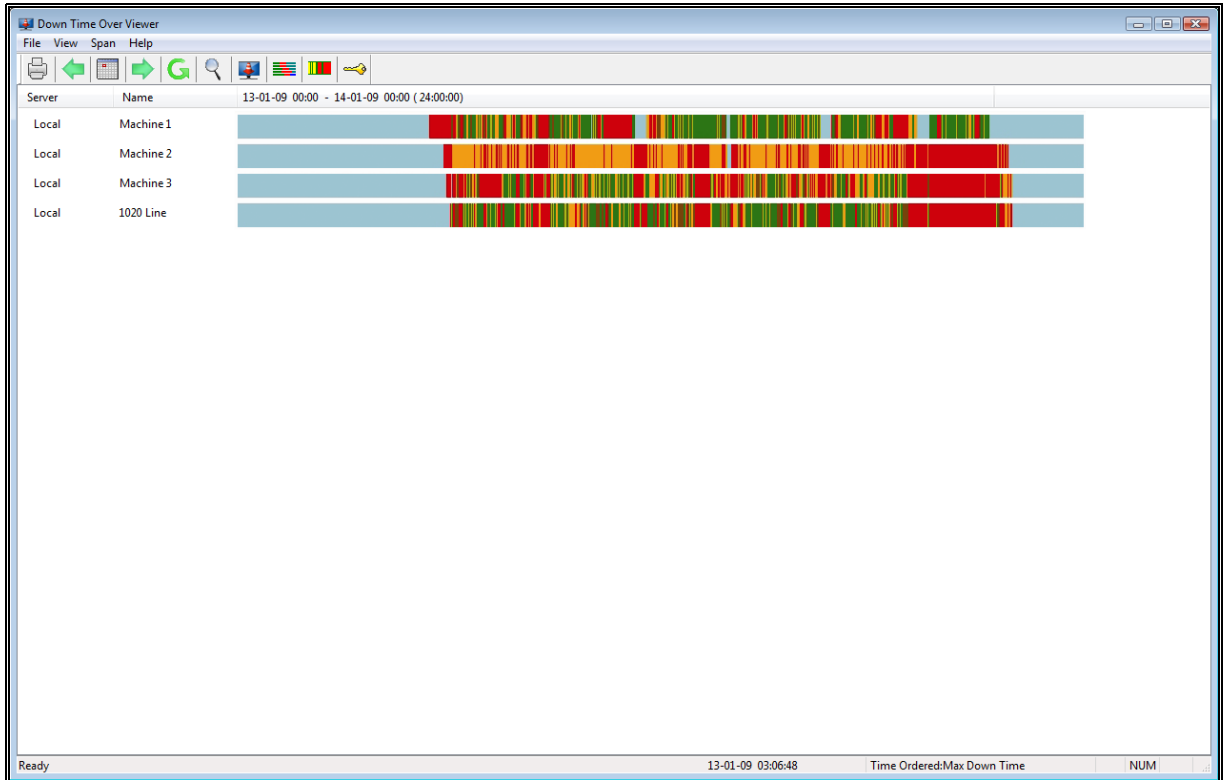
The Date/Time Interval Size allows the user to select the time period over which total downtime will be grouped for representation in a report. If the data you wish to view is over two days and the Interval Size is 1d as above there will be two records in the produced report, one for each day; if the Interval Size is 12h there will be four records in the produced report.

If the Include check box in the PNTR section is checked plan not to run downtime data will be included in the report, this may be misleading as PNTR data might include overnight when a machine wasn't running so the largest reason for downtime might be "not manned" and this would overshadow other information in a report.

To run the report, click the **Run Report** button. Down Time Viewer will produce a database containing the relevant data, sorted accordingly, and then fires up the selected report.

Down Time Overviewer

The Down Time Overviewer allows you to view down time data from many machine centres simultaneously as a time based horizontal bar chart on the screen. The data is represented by different coloured bars along a time period and lends itself to easy comparison between machines to identify similar periods of downtime. The Down Time Overviewer screen is shown below and the use of the program is described in full in the following section.



The screen has a toolbar along the top that controls how the data is viewed. Below that is the main display area of the program in which all of the down time data is displayed.

The Tool Bar

The toolbar can be seen below; it consists of a number buttons that allow control of Down Time Overviewer each of the buttons is described below:



- **Print.**



This will print the downtime data that can be seen in the main window, it will use the same time base and display a summary of any options that are set such as Time ordered/Downtime ordered or Maximum/Average.

- **Step Back.**



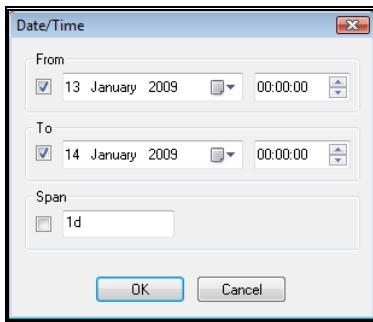
This will step back one time period, so if the time period is one day this will step back one day in time.

- **Select Date.**



This button displays a dialog that allows you to select a start time, end time or span around which

you want to view the downtime data. The dialog that you select the time period from can be seen below.



- **Step Forward.**



This will step forward one time period, so if the time period is one day this will step forward one day in time.

- **Refresh.**



This causes the Down Time Overviewer to re-read the downtime data. New events may have occurred since the Overviewer was opened, and this will cause them to be displayed (provided they occurred within the timeframe currently being viewed of course).

- **Zoom.**



This allows you to zoom in on a specific period of downtime to see the reasons why the downtime occurred.

- **Down Time Viewer.**

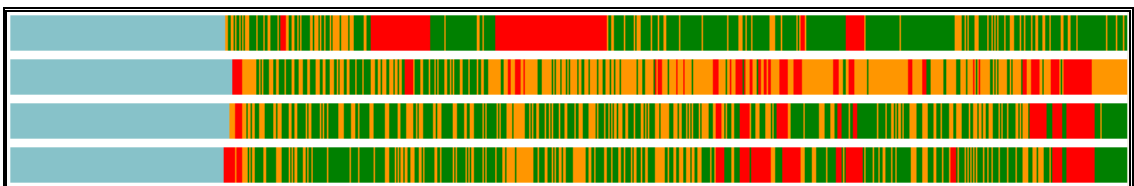


This button will run up the Down Time Viewer program that was described earlier.

- **Time Ordered/Downtime Ordered.**



Below is downtime data that is Time ordered.



Pressing the Time Ordered/Downtime Ordered button will change the data so that all the Uptime, Downtime etc. are grouped together. This makes it a lot easier to compare the amount of downtime/uptime/short stoppages on each machine.

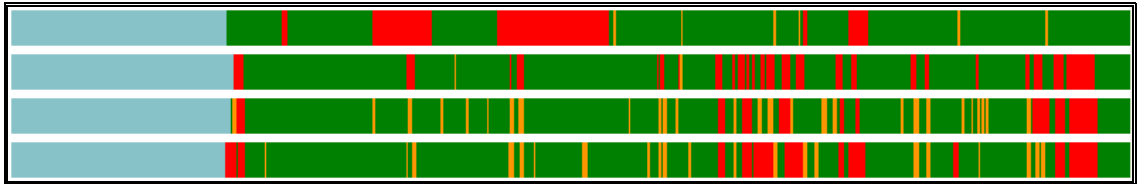


- **Maximum/Average.**



Pressing the Maximum/Average button will change how Down Time Overviewer displays the data, in some cases the time period is too great and there are not enough pixels on the screen to display every downtime and uptime event.

The default view for Down Time Overviewer is Maximum mode. In this mode if there is any downtime during the period of time that a pixel represents then that pixel will be labelled as downtime. This ensures that any downtime is shown on the screen (it could be said that the display is downtime biased). When the Maximum/Average button is pressed the display will change between the two modes, Average mode for the example data is shown below.

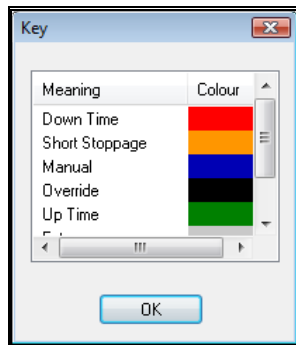


In Average mode the data that a pixel will represent is analysed in a different way, if **over** half the time the pixel represents is uptime then the pixel will be coloured green and will represent up time. If less than half the time a pixel represents is uptime then the pixel will be coloured as the appropriate non up time occurrence; e.g. Downtime, Short Stoppages, Manual, Override etc.

- **Key.**

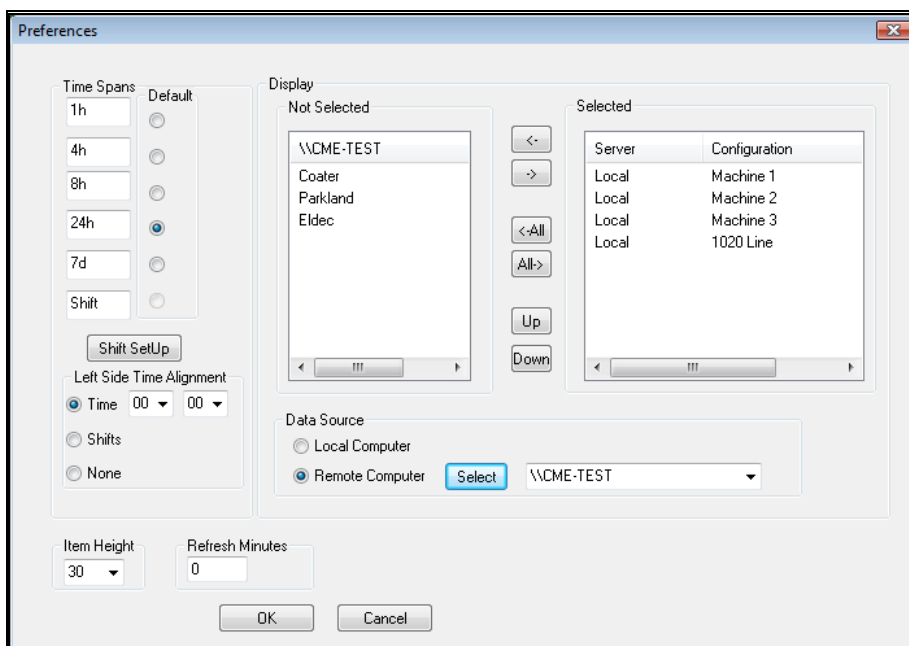


This loads a key up that shows the user what each of the colours on the screen represents. The dialog that is loaded can be seen below.



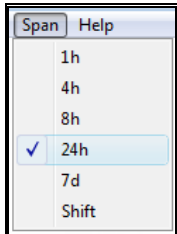
Preferences

The preferences dialog can be reached by clicking on File followed by preferences on the main menu bar. The dialog that is displayed can be seen below and it allows configuration of Down Time Overviewer.



Time Spans

On the main menu bar there is an option called **Span**; this menu contains the values that can be seen within the **Time Spans** section of the Preferences dialog. Changing the values in this section of the preferences dialog will change the values that are displayed on the main menu bar. The default check button allows the selection of the time period to be used when Down Time Overviewer is run. The Span Menu can be seen below.



The **Left Side Time Alignment** determines the time that will be used as a starting point on the left hand side of the bar. Change the time by entering in a time value into the drop down boxes. The default value is 00:00.

Display

This section allows you to configure which machines will have their down time data shown in the Overviewer main window. As a default all of the machines on the local system are selected.

Single machines can be removed or added (Selected or Not Selected) by selecting a machine in either of the lists and clicking either move right **→** or move left **←**. This has the effect of moving the machine to the other list. All machines can be removed or added by clicking either All **→** or **←**All. The position of a machine in the selected list can be changed by selecting the machine that you wish to move and then pressing the Up or Down buttons.

Down Time Data can be viewed from machines that are on the local Prodigy system but also from remote machines that have downtime systems on them and which can be seen on the network. The **Data Source** section allows a remote computer to be selected; to do this simply check the **Remote Computer** button and then press the **Select** button. This will bring up a standard windows dialog used to browse for files; find the machines that are available on the network that have Prodigy systems on them and point the file browser to the Tascomp Ltd folder (which must be shared) on these machines. If there is downtime data on the system that you selected the appropriate machines will be made available for selection in the **Not Selected** area. They can then be selected in the same way as local machines.

Multiple remote systems can have their data selected simply by repeating the above steps until all required machines have been added to the **Selected** list.

Item Height

This allows the height of the horizontal bars that display the downtime to be configured; the value can be selected from a drop down list and it is the number of pixels the bar is high.

Refresh Minutes

This value can be entered to make Down Time Overviewer automatically refresh its data every X number of minutes.

Data Processing

Data Processing operates on Down Time and Batch Data to produce a unified output file that is suitable for reporting on using Easy Report or some other reporting tool.

Data can be collected from different batch and down time databases in different locations and accumulated into one database that contains all of the tables and fields that are necessary for a particular style of report. Exactly how the data is accumulated and in what format the output database appears depends upon the data processing plug-in that is selected to do the work. Prodigy comes with a number of standard plug-ins and customer-specific ones can be provided as part of a project or at additional cost to cover any specific needs that arise.

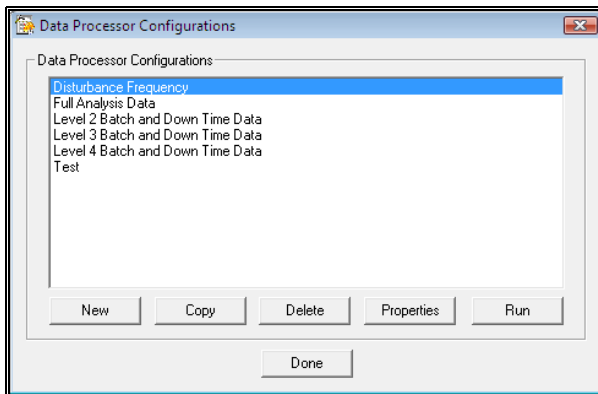
When configuring a down time system that will monitor the status of several discrete lines, processes or machines, you should bear the following in mind.

- Configure Batch Data Recorder so that the batch data for each process is recorded in a different batch database (one per process).
- Configure the Down Time Logger so that the downtime data for each process is recorded into separate directories (again, one per process).

In this way you will be able to tie together batch and downtime data that belong to the same process when you run any data processing.

Data Processor Configuration

Configurations

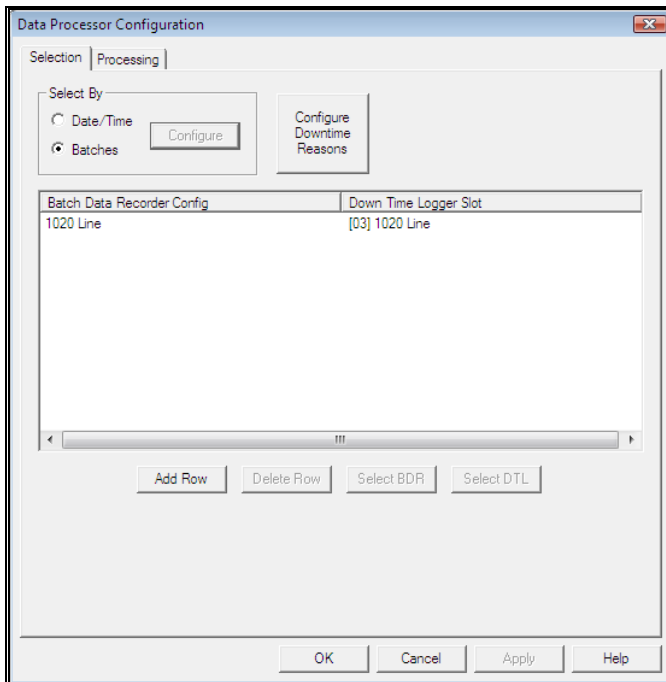


The Data Processor Configurations screen shows you the existing configuration names and allows you to :

- Create new configurations.
- Delete existing configurations.
- Make copies of existing configurations.
- Modify existing configurations.
- Run an existing configuration to produce an output database.

When creating or copying configurations, the name that is given must be unique otherwise an error dialog will be displayed.

Properties



There are two tabs on the Data Processor Configuration dialog, Selection and Processing.

Selection lets you decide how data is selected for analysis, either by Date/Time or by Batch(es) and it also lets you decide where to look for the data. Finally it allows you to specify which Downtime reasons at each level are to be considered.

The Processing tab merely allows you to specify which data processor to use and where the output data is to be placed.

Selection Tab

Batch Data/Down Time Data Selection

Begin by selecting the area (or list of areas) from which data will be taken. Add a row of data selection information by clicking the **Add Row** button, which will create an empty row in the list (both sides will read <none>).

You can specify where Batch Data and Down Time Data will come from by clicking the **Select BDR** and/or **Select DTL** buttons. Clicking one of those buttons will allow you to choose a configuration for that aspect of the system, which in turn contains the location of either the Batch or Down Time data, as appropriate.

When selecting pairs of configurations you should ensure that they relate to the same aspect of the plant, e.g. the same process, line or machine. There are some rules that must be adhered to when creating the list of Batch/Down Time pairs:

- A row with both items set to <none> is meaningless and is not allowed.
- You do not have to have any batch data, you may if required select only down time data.
- You do not have to have any down time data, you may if required select only batch data.
- If you select batch data in one row, you must select it in every row.
- If you select downtime data in one row, you must select it in every row.
- You can select batch data by batch name if you have one and only one row. If you have multiple rows batch selection will always be by date (all batches that fall within the specified date range will be selected).

Select By

When the data processor runs it will ask the user to select the period for which the down time data will be analysed. This can be done in one of two ways, either by date/time or by batch, with the further option that a range of batches may be selected by entering the date/time period in which the batches must fall.

The type of selection that is allowed depends upon what is in the columns described earlier.

- If there is any Batch Data selected then selection is always by batch.
- If there is more than one row with Batch Data specified then selection is always by batches within a specified timeframe.
- If no Batch Data is specified then data selection is always by time periods.
- If there is only one row containing Batch Data then selection is optionally by batch name or by batches within a timeframe.

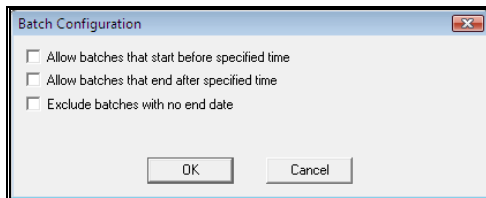
When Batch Data Is Specified

When selection is by batches (either explicitly or by batches within a timeframe) the data will be processed on a batch by batch basis. For instance if you select two batches that ran today, one from 8:00 till 12:00 and the other from 13:00 to 18:00 then only downtime that occurs within those two time periods will be analysed. Anything that occurs, for instance, between 12:00 and 13:00 will be ignored, as will anything outside of the two batch periods. Similarly uptime is calculated based on the periods that the batches were running.

When selecting batches by time frame you can specify how batches that overlap the time period are handled. For instance a batch may start before the specified timeframe and finish within it, or start within it and finish outside of it. It may even start before and end after, overlapping the timeframe entirely.

Additionally, a batch may have no end date for some reason, the most likely of these being that it hasn't yet finished (or, to a lesser degree, that something went wrong with the system before the batch ended). If you are including batches that start before the specified time, then if a batch has no end date it will be assumed that it is still running and therefore overlaps the period specified. You can optionally exclude batches with no end date, or (the default) include them.

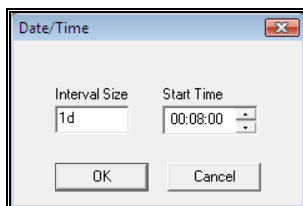
To configure how these scenarios are treated click the **Configure** button.



In addition, when Batch Data is specified all of the fields from the batch data record are added to the downtime record for that batch in the output database. This means that for each batch record that matches the selection criteria specified you will end up with a matching record in the output database that combines the downtime and batch data for that batch.

When No Batch Data Is Specified

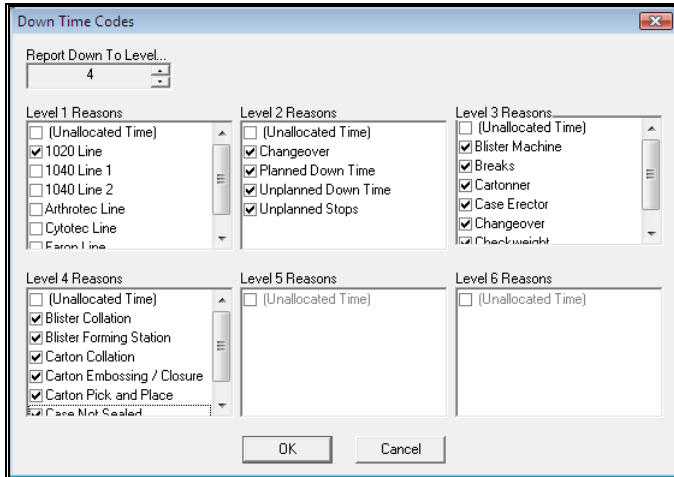
When no Batch Data is specified only selection by date/time is allowed and **all** down time events that occur within the specified timeframe are analysed. The data is divided into time slices whose duration are specified by clicking the **Configure** button.



In addition you specify the start time for the slices. In the example shown a new slice will begin at 8:00 each day. If the Interval Size were 8 hours (8h) a new slice would begin at 8:00, 16:00 and 0:00 each day and so on.

Configure Downtime Reasons

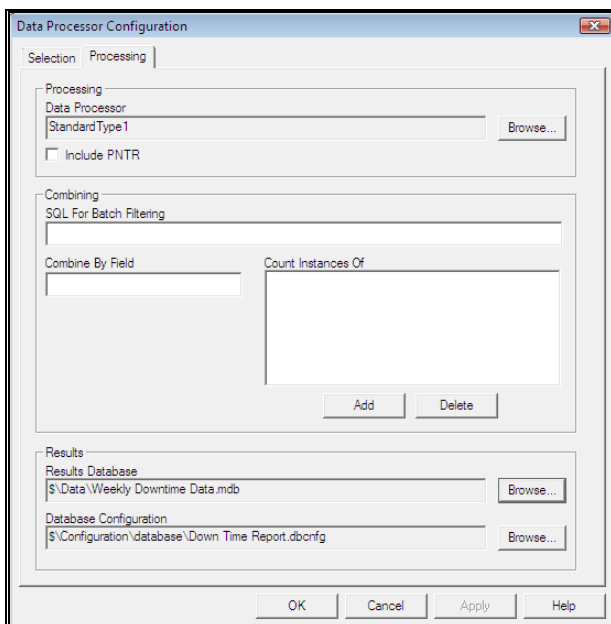
You **must** select the level that data is to be accumulated at and within each level up to and including that level you can select which downtime events to include. The screen is as follows:



Specify the level at which data is to be accumulated. This means that in the example above the data will show how much downtime was allocated to the level 4 downtime reasons, **Blister Collation**, **Blister Forming Station**, **Carton Collation** etc. If we had selected a level 3 report we would see how much downtime was allocated to **Blister Machine**, **Breaks**, **Cartonner** etc. The lower the reporting level, the more detail you will see.

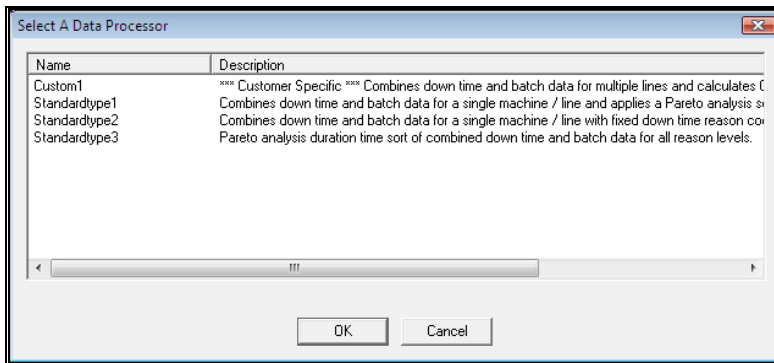
As you check or clear boxes at one level to indicate whether or not you are interested that particular reason, the reasons shown in the subsequent boxes may change. This is because each successive Reasons box shows only those reasons that are subsets of the reasons selected at a higher level.

Processing Tab



The Processing tab allows you to specify which Data Processor to use and where the resulting database is to go. Data Processors are small plug-in modules that process the data in a specific way according to the type of report that is required. A number of standard plug-ins are provided as are a number of customised “special” processors that are written to meet customer’s specific requirements. Additional processors may be written from time to time as needs arise. A list of available processors, together with a brief description of what they do, is displayed when the **Browse** button is clicked.

The processor selection dialog is shown below and contains three standard processors and one custom one. The function of the standard processors is described later.



You should specify the location and name of the output database that will be produced by the data processor (it will default to a results database in the Data folder). Do this by clicking the **Browse** button next to the relevant field. In addition you may need to alter the default setting for the Database Configuration file for the results database (this is created and/or modified using Access Database Configuration, described elsewhere in this manual).

The processing dialog also allows you to specify any *data combining* that may be required. Data combining combines records that share a common attribute value into one record. For instance, if you select 30 batches and combine by “Product Code” you will end up with one record per unique product code (so if there are 5 different product codes used throughout the 30 batches, you will get 5 records).

For each combination record all numeric data from the individual batch records is totalled up, field by field. Text data is set to the value of the first non-blank value that is encountered as the batches are combined. Time data is set to show the earliest and latest dates that occur within the records that were combined. This is achieved by finding the earliest value of any time field containing the word “Start” and the latest value for any field containing the word “End” (e.g. Start Date, End Date, Batch Start, Batch End and so on).

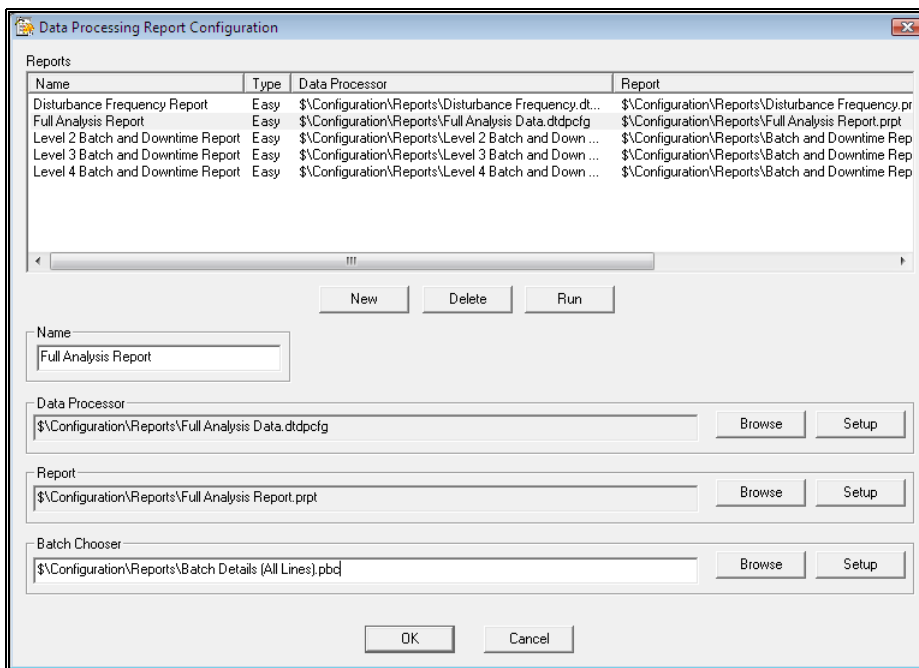
The total number of batches that were processed into the combined record is also recorded.

Not all numeric data should be added together by the combining process, for example serial numbers. To prevent this you should enter the names of such fields into the “Count Instances Of” box. What will happen then is that instead of totalling up the fields it will count the number of unique instances that occurred within that field. For example, if there were 11 different serial numbers used within the 30 batches that were combined into this record, there will be a field called “Count Of Serial Number” which will contain the value 11. You can do this with any field, not just numeric fields, to find its count of unique instances and to prevent any meaningless totalling.

Down Time Reporting

Once data has been collected and pre-processed you can use it in your report. Reports can be produced by several utilities within Prodigy but the most useful for Down Time reporting is *Easy Report* (see elsewhere). A program exists within Prodigy’s Down Time suite that allows you to tie a particular report to a particular pre-processor and to run this via a button on the *Menu Bar* or a *Display Builder* screen.

Data Processing / Report Configuration



- Add or delete report/pre-processor pairings by using the New and Delete buttons.
- Run the selected report by clicking the Run button.
- Enter a name for this report in the Name field.
- Select the required Data Processor configuration (described earlier), Report and (if required) Batch **Chooser** by clicking the relevant **Browse** button. The **Setup** button alongside these will run up the relevant program to allow you to check or create the configuration to be used. A Batch Chooser configuration is not mandatory. If used, a Batch Chooser configuration allows you to specify which fields are shown in a Batch Chooser window and also specifies a selection criteria that allows you to pre-filter the batch records.

Running A Down Time Report

To run the Down Time report/processor pairings configured above, use the Data Processed Reports (DTRunReport.exe) program from either the Menu Bar or from a button on a Display Builder screen. In addition to the name of the configuration that is to be run, the program takes any or all of the following arguments, which can be used to override the pre-configured options.

- **-name**
The name of the processor/report pairing, configured as described above.
- **-filter**
The path & name of the Data Processor Configuration to be used in place of the one that was pre-configured.
- **-report**
The path & name of the report to be used in place of the one that was pre-configured.
- **-database**
The path & name of the database to be used in place of the one that was pre-configured.
- **-chooser**
The path & name of the batch chooser configuration to be used in place of the one that was pre-configured.

Using different parameters will allow you to use the same basic configuration but called with a different **-filter** option to produce output for different lines or to different levels and so on.

Shop Floor Terminals

What Are Shop Floor Terminals?

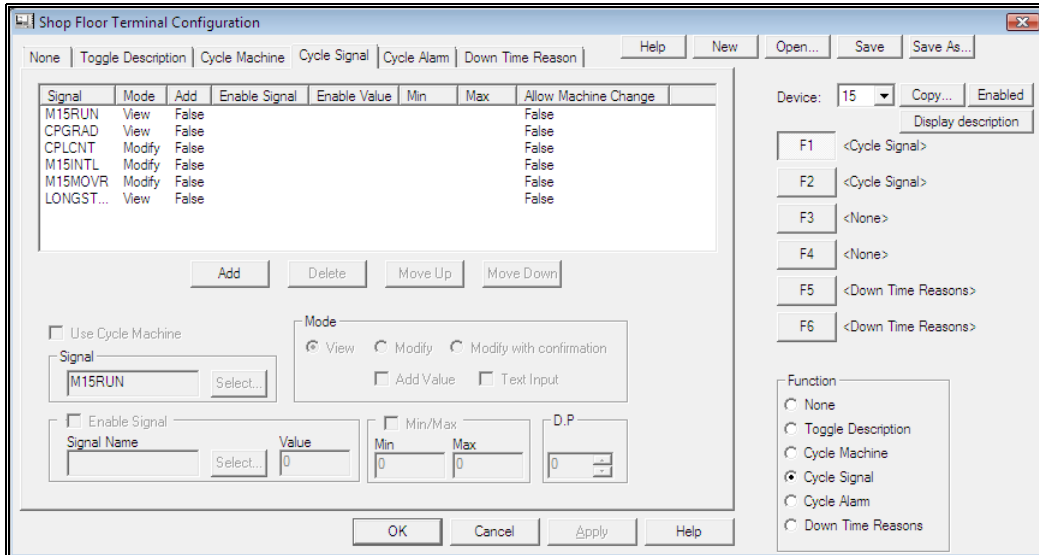
Shop Floor Terminals are fully sealed units that can be used on the shop floor as an alternative to a remote PC workstation. They have a simple numeric and function key approach that can be configured to allow interactions with a Prodigy system. Examples of such terminals are Burr Brown and Tascomp's own range such as the TM01, DTM02 and DTM03 (shown below). These latter terminals are optionally available with digital I/O, barcode readers and T9 text input.



Any signal in the system can be viewed on the LCD display and new values for defined signals can be entered. The terminal can also be configured to provide other functionality including the viewing and accepting of current alarms.

Shop Floor Terminal Configuration Program

Up to 32 Shop Floor terminals can be put on the same RS-485 multi-drop communications loop. For each loop in the system there must be a separate Shop Floor Terminal driver, each with its own configuration file which is stored in a standard area within the Prodigy directory structure. The Shop Floor Terminal Configuration program presents a list of the configuration files present in this area when the **Open...** button is pressed.



- **Help.**
This button displays help about Shop Floor Terminal Configuration.
- **New.**
This button begins a new configuration.
- **Open...**
This button opens an existing configuration. A standard file-chooser dialog will be displayed.
- **Save.**
This button saves the current configuration.
- **Save As.**
This button saves the current configuration under a new name.
- **Device.**
Each Shop Floor terminal on a given loop must have a unique device address and this is then used to determine its configuration. Selecting the required Shop Floor Terminal address is simply a matter of choosing it from the **Device** list, noting that numbers with an asterisk are currently unused.
- **Copy.**
When one *Device* has been fully configured, this configuration can then be copied to another *Device* by pressing the **Copy...** key and selecting the device to copy to.
- **Enabled/Disabled.**
Individual terminals can be temporarily removed from the loop through use of the **Enabled/Disabled** option. Note that the button shows the **current** state, not the state that pressing the button will apply.
- **Display Description/Display name.**
Each device can also be configured to show either the signal descriptions alongside their values or the signal tagnames. Do this by pressing the **Display description** button, which then toggles to show **Display name**.
- **F1-F6.**
These represent the six function keys of the shop floor terminal. Click on the function key that you want to configure and then select the relevant **Function** for that key.

- **Function.**
 - **None.**
The function key performs no action.
 - **Toggle Description.**
The function key toggles the shop floor terminal display between signal name and signal description.
 - **Cycle Machine.**
The function key switches the machine to which actions on this terminal apply.
 - **Cycle Signal.**
The function key cycles around the list of signals that are defined for this shop floor terminal.
 - **Cycle Alarm.**
The function key cycles around the list of alarms that exist for the signals defined for this shop floor terminal.
 - **Downtime Reason.**
The function key allows viewing and/or entry of down time reasons.

These options are all described in more detail below.

Device Configuration

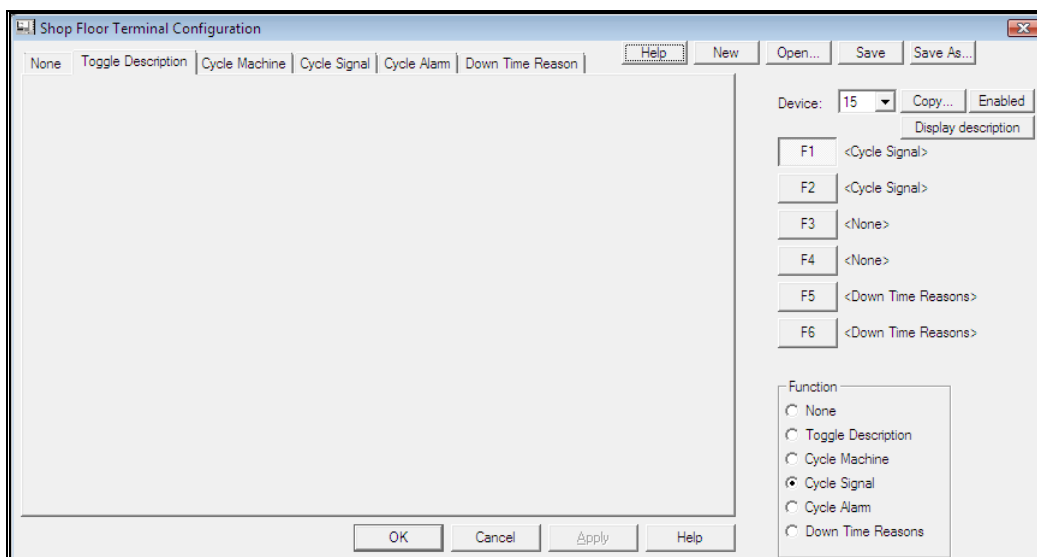
With the required device selected, the six function keys, F1 to F6, can be assigned one of five actions. Click on the function key that you want to configure and then select the required action from the list of radio-buttons below. The correct tab for configuring that function will be displayed. *Do not change to a different tab by clicking on the tab itself, as this is meaningless.*

The various functions available are described below:

None.

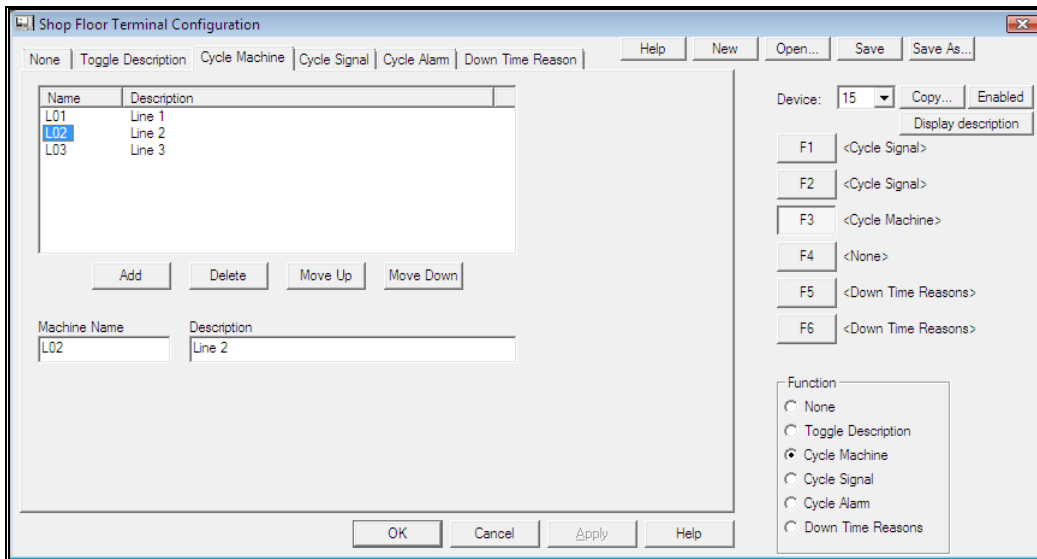
This sets the function key to be unassigned.

Toggle Description.



When this function key is pressed, the display will show signal descriptions instead of signal tagnames, or vice versa.

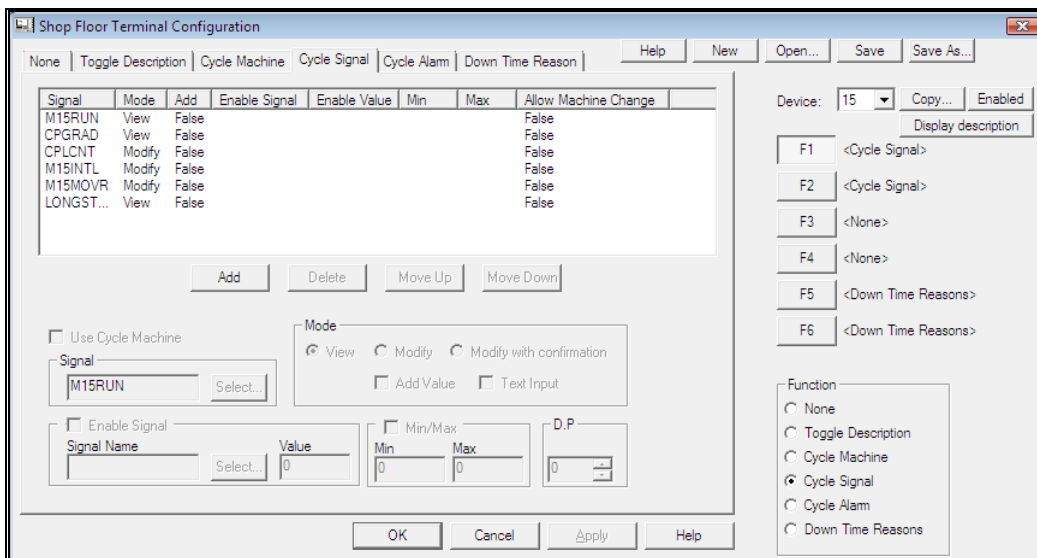
Cycle Machine.



This function is used to allow one Shop Floor terminal to be used on a number of pieces of equipment and simply cycles through the available machine names. Each time the function key is pressed the display will show the next one of the **Descriptions** defined above and the substitution string will be set to that shown in the **Machine Name** field.

When other function keys are subsequently pressed the characters in the **Machine Name** field will be used to replace the same number of leading characters in the tagnames assigned to those other keys. In the Cycle Signal example shown below the tagnames that begin with L01 will cycle between L01, L02 and L03, as defined above.

Cycle Signal.



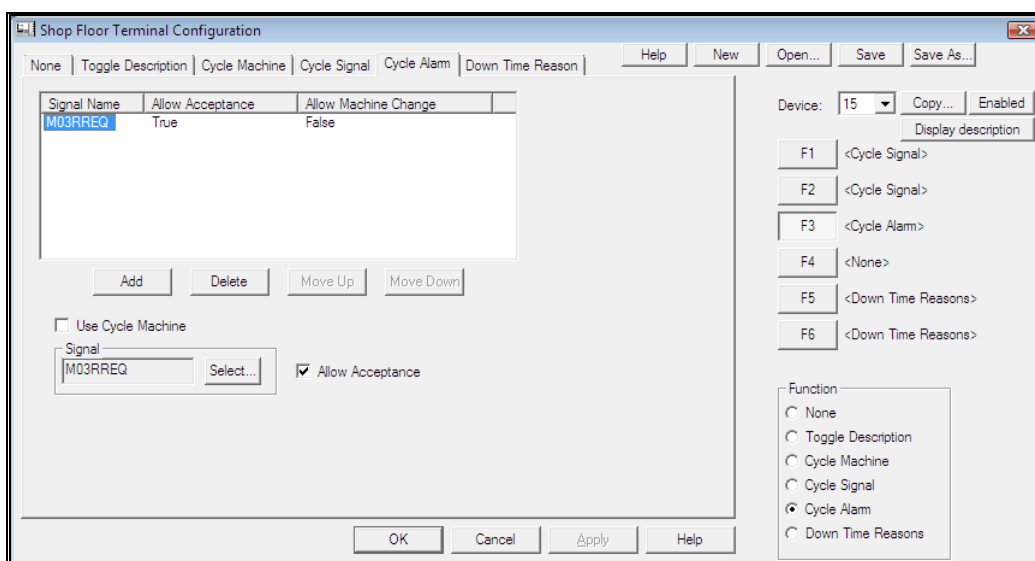
This function is used to cycle through the available signals to either view the current values or to allow new values to be entered.

- **Buttons.**
- **Add.**

Any number of signals can be added into the list by pressing the **Add** button and selecting the signal require.

- **Delete.**
Conversely, removal of a signal from the list is achieved by selecting it and pressing the **Delete** key.
- **Move Up/Move Down.**
The order of the list may be altered through use of the **Move Up** and **Move Down** keys.
- **Mode.**
 - **View.**
This is a read-only mode. The selected signal values can be viewed on the shop floor terminal but cannot be modified.
 - **Modify.**
The signal values can be modified.
 - **Modify with confirmation.**
Signal values can be modified but an additional confirmation is required before the change is applied.
 - **Add Value.**
The value entered at the terminal is *added* to the current value of the signal. If this box is not checked the value entered at the terminal *replaces* the current value of the signal.
 - **Text Input.**
If this is checked the value entered at the terminal is placed into the *Event Text* field of the specified signal (it is placed in event text position 1).
- **Enable Signal.**
If this is checked then changes to the specified signal are only allowed when the enabling signal matches the specified value.
- **Min/Max.**
Values entered at the terminal are validated against the range of the signal as specified in *Signal Database Configuration*. You can override this by checking the Min/Max box and specifying new Min and Max values.
- **DP.**
This specifies the number of decimal places to use when displaying values at the terminal.
- **Allow Machine Change.**
This option is used to determine if the **Cycle Machine** facility can be applied to these signals.

Cycle Alarm.

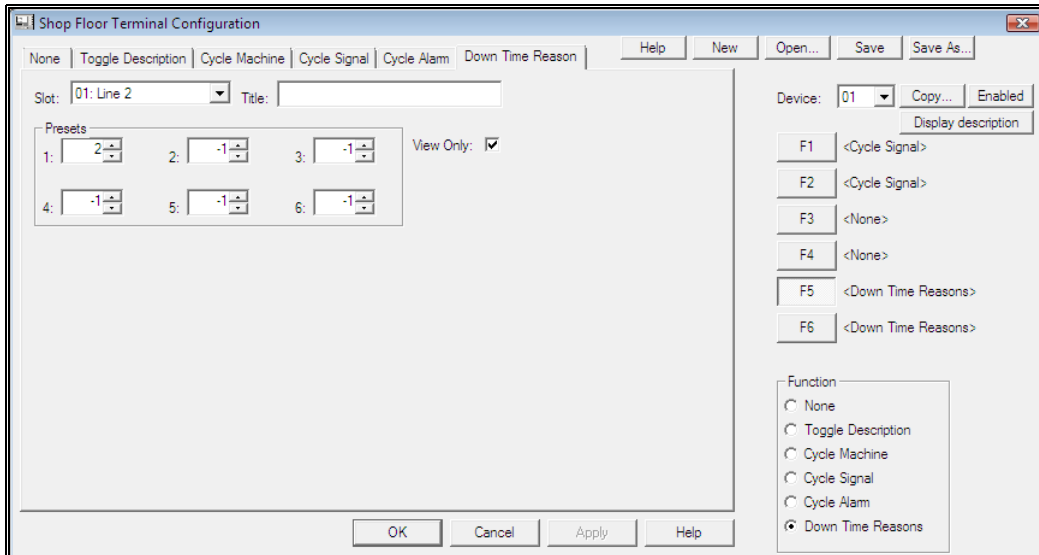


This function is used to cycle through the current alarms.

Any number of signals can be added into the list in the same manner as used for the **Cycle Signal** facility.

- **Allow Machine Change.**
This option is used to determine if the **Cycle Machine** facility can apply to these signals.
- **Allow Acceptance.**
This tick box is used to allow acceptance of any alarms on this signal via the Shop Floor terminal.

Down Time Reason



This function is used to allow down time reason codes to be viewed and/or entered. Choose one of the down time configurations that have been previously created using the Down Time Logger Configuration program.

If the Title field is filled in, this will be displayed on the top line of the terminal. It is recommended to only use this when you have several function keys on one terminal set up to display down time from different configurations, as a side effect is that there is one less line of display available for showing down time information.

Presets can be chosen for the function key. This allows the down time reason choices to start at a level other than the top. For example assume that the Level 1 is the line number (Line 1, Line 2 etc) and that Level 2 is the type of down time (mechanical, electrical etc). A preset Level 1 could be applied so that the downtime reasons displayed on the terminal begin at Level 2 (mechanical, electrical etc), and always apply to Line 1. A value of -1 means no preset.

The function key can be configured to only show the current state of down time and not allow it to be changed by ticking the View Only box.

SLANG (Sequence LANGUAGE)

What Is SLANG?

SLANG is a simple programming language with which Prodigy users can write control programs or routines. The language has been designed to be as simple as possible, having few keywords and statement types and making use of an extensible runtime library of external functions.

The SLANG environment consists of three distinct parts.

- The **SLANG Editor**.
This is a context sensitive editing environment that allows the creation and modification of SLANG programs that are guaranteed to be syntactically correct.
- The **SLANG Runtime**.
This is (optionally) started as part of the Prodigy services and is responsible for executing SLANG programs as and when required.
- The **SLANG Control Panel**, which provides a user, interface to the runtime environment.

The SLANG Language

Rather than describe the language in formal language definition terms, the SLANG language is described below in simple English.

Variables

Variable names are case-insensitive. Whichever way you enter a variable name (or a macro), SLANG will interpret it as though it was all upper case. The name can consist of any characters from A to Z, any digits from 0 to 9 and the underscore character (_).

Types

- Two **types** of variables are allowed - **Numeric** and **String**.
 - Numeric and String variables may each have names of up to 30 characters.
 - Numeric variables all hold floating-point real numbers.
 - String variables are identified by having a % as their final character.

Example

VAR1 – Numeric variable.
VAR1% - String variable.

Note, the above example is valid, in that the % character is used as part of the name so VAR1 and VAR1% are recognised as two different variables.

- In addition, **arrays** of variables are allowed too.
 - Only one dimensional arrays are allowed, and an array is specified by the use of [] brackets. The statement **set A[2] = 3** sets array element **2** of the array **A** to the value of **3**.
Note that arrays start with index 0, as in the C and C++ languages. Arrays grow dynamically and can be “sparsely” filled – our example used above automatically sets the array to have 3 elements, and elements 0 and 1 are set to contain zero until explicitly set. A further statement **set A[10] = 6** will increase the array size to 11 (remember it’s zero based) and elements 3 to 9 will contain zero.
 - You can set the upper bound of the array by using a statement of the type **set a[] = 20** which will set the upper array bound to 20. Any statement that tries to access an array element beyond that will fail and will return zero if used as an expression. It will also result in a message being output to *Spy* saying what the attempted subscript was and what the maximum allowable subscript is.

- You detect the upper bound used by using a statement of the form **set upperbound = a[]**. Upperbound (which is simply a variable name, rather than a reserved word) will contain the upper bound limit of the array.
- The subscripts of an array can be expressions, as in **set a[3 * I - 2] = jlk * 23**.
- String variables can also be arrays – **set abc%[1] = “abc”**.

Scope

There are three **scopes** of variables, **Local**, **Global** and **Signal**.

- Local variables are unique to the SLANG program in which they occur. If there are two programs, each may have a local variable called ABC and the value held in this variable will be different for each program
- Global variables are identified by having a ^ as their first character. The same variable is used throughout any and all SLANG programs. If two programs use the global variable ^ABC any changes to the value made by either program will be seen by the other.
 - ^ABC is a global real variable.
 - ^ABC% is a global string variable.
 - ^ABC[2] is element 2 of the global array ABC.
- Signals.
 - Signals are identified by having a \$ as their first character.
 - Signal names may only be 8 characters in length, excluding the \$.
 - Signals do not have a % at the end to represent it is a string.
 - You cannot have an array of signals.
 - Assigning values to a Signal will have the effect of modifying the signal's value so that every part of Prodigy that references that signal will see its change of value (e.g. Display Builder, Alarms, BDR, SLANG). By definition, Signal variables are global. Numeric signals (i.e. real, integer, digital and finite state) can only be set to numeric values and string signals may only be set to string values. An error will be displayed if you attempt to set the signal to the wrong type of value.

Examples

VAR1 – Local numeric variable.

^VAR2 – Global numeric variable

VAR3% - Local string variable.

^VAR4% - Global string variable.

\$AI01 – Signal variable (by definition this is global).

Indirection

A mechanism is in place for allowing variable indirection with Signal variables. To do this construct a variable name beginning with the & character (e.g. &MYVAR). This is an indirection variable and behaves in exactly the same way as a signal variable except that you can change its name by using the NAME attribute (e.g. &MYVAR.NAME = "SIGNAL1"). This allows you to write a routine that can operate on a range of different signals, for example.

```
SET SIGNAME%[0] = "AI01"
SET SIGNAME%[1] = "AI02"
SET SIGNAME%[2] = "AI03"
SET SIGNAME%[3] = "AI04"
.
.
SET LOOP = 0
REPEAT UNTIL LOOP = 4
  SET &MYVAR.NAME = SIGNAME[LOOP]
  IF (&MYVAR > 100) THEN
    SET &MYVAR = 0
  ENDIF
.
.
  SET &MYVAR.DESCRPTION = "Processed"
.
.

  LOOP = LOOP + 1
ENDREPEAT
.
.
```

Indirect variables are always of local scope, though of course the signals that they represent are of system-wide scope.

Constants

Only two types of constants are allowed, **numeric** and **string**.

- A numeric constant is simply any number as you would normally type it.

Examples

```
1
123
123.45
```

- A string constant is any characters enclosed in double quotes ("). If you want a " to appear in the string, precede it with the \ character.

Examples

```
"Hello"
"This is a string"
"My name is \"Peter\""
```

Macros

There are several macros available within SLANG. The purpose of a macro is to make the program more readable by providing a plain English equivalent of a constant data item or function call. The macros available at present are:

- **YES, ON, OPEN, RUNNING, TRUE.**
These all equate to 1 and may be used freely to replace the integer constant 1.

- **NO, OFF, CLOSED, STOPPED, FALSE.**
These all equate to 0 and may be used freely to replace the integer constant 0.
- **DISABLED, ENABLED, UNKNOWN.**
These refer to signal states. A **Disabled** signal is one that has been made temporarily unavailable to the system (maybe due to maintenance or hardware failure, see the **Signal Disabler** section). **Enable** can be used to re-enable a Disabled signal. An **Unknown** signal is one that has no current value. This is usually due to hardware or other system faults.
- **TIME.**
This equates to a real number representing the current date and time, as used throughout Prodigy.
- **DBBOOL, DBDATE, DBDOUBLE, DBINT, DBMEMO, DBSTRING.**
These relate to SLANG database extensions and represent various data types available within databases.
- **DBFIRST, DBLAST, DBNEXT, DBPREV.**
These relate to SLANG database extensions and represent record navigation keywords.
- **ARGCOUNT.**
This special macro can only be used within a routine and returns the number of arguments that were passed to that routine.

Examples

```
set A = Off
if ($DI01 = Closed) then
  Set Now = Time
else
  if ($AI01 = unknown) then
    .
    .
  endif
endif
```

Statements

Set

Set variable = expression

where **variable** is as explained above and **expression** is either a constant or a mathematical formula. The variable may be of numeric or string type. In either case the expression must consist solely of variables and/or constants of the same type, and the only mathematical operator allowed in a string expression is +. Expressions are explained fully later in this section.

Examples

```
Set A = 12
Set B = 3 * A
Set $AI01 = (A + B) * 12.5
Set D% = "This is a string"
Set E% = D% + " expression"
```

Turn

Turn variable On

Turn variable Off

Turn is essentially the same as Set but it is tailored towards digital variables.

Lock, Unlock

Lock variable

Unlock variable

The Lock and Unlock statements are used to ensure mutual exclusion. If one program locks a variable and then another program tries to lock the same variable, the second program will be forced to wait (i.e. it will stop executing) until the variable is *Unlocked*. In order for locking to work, the variable being locked must be global. This means that Signal variables can be used for locking purposes.

Notes.

Any program can unlock a variable that has been locked, it does not have to be unlocked by the same program that locked it.

Be aware that locking a variable does not mean that its contents cannot be altered by another program. Locking a variable only affects other Lock statements, and not things such as Set or Turn. Good programming practice would be to use a particular variable as a semaphore **and only as a semaphore**.

Wait

Wait Until condition

Wait For expression

The first form of the Wait statement *Wait Until condition* means that the next line of the program will not be executed until the specified condition is satisfied. A condition is a form of expression based on relationships that returns 1 or 0, depending on whether the relationships match those specified (conditions are explained in full later in this section).

The second form of Wait statement *Wait for expression* means that the next line of the program will not be executed until the time (in seconds) that **expression** evaluates to has elapsed.

Examples

```
Wait Until A > B
Wait Until $DI01 = Off
Wait Until A = B Or C = D
Wait For 25
Wait For A * 3
```

If Else Endif

If condition then

.

Else

.

Endif

Or

If condition then

.

Endif

The If – Else – Endif block provides a way of doing one thing or another depending on the value of the condition. If the condition is true then the line of program following the If is executed. If the condition is false the line of code following the Else (or the Endif, if there is no Else) is executed. When execution eventually reaches the Else statement, execution continues with the first line of code following the Endif.

Examples

To set C equal to the greater of A or B

```
If A > B Then
Set C = A
Else
Set C = B
```

Endif

To ensure A is positive

```
If A < 0 Then
Set A = -A
Endif
```

To compare strings

```
If A% = B%
If A% = "String"
```

Repeat

Repeat Until condition

.

EndRepeat

Or

Repeat expression

.

EndRepeat

Or

Repeat forever

.

EndRepeat

In the first form of this statement, the statements falling between the Repeat and EndRepeat are executed until **condition** is true. In the second form, the statements are repeated for the number of times that **expression** evaluates to. Note that in both cases it is possible that the statements between Repeat – EndRepeat will never be executed (if **condition** is initially false or **expression** evaluates to 0 or less). Note also that with the second form of the statement the **expression** is only evaluated once and thereafter the value that was arrived at is decremented with each successive cycle of loop.

The third form of the repeat continues forever. If you wish to exit the repeat loop you must use a *goto* to jump out.

When writing loops in SLANG (by any method) be aware of a condition known as a “tight loop”. This is where code loops frequently and rapidly and contains no statements that will cause it to “pause”. SLANG will execute these statements as rapidly as it can, to the detriment of other processes. In order to avoid this situation, or to cure it if you see that it is happening, put a “WAIT FOR n” statement just before the EndRepeat n can be very small usually, say 0.01.

Examples

Both of these examples set a variable to 1 and then increment it up to 11.

```
Set A = 1
Repeat Until A > 10
    Set A = A + 1
    Wait For 0.01
EndRepeat
```

```
Set A = 1
Repeat 3 + 7
    Set A = A + 1
    Wait For 0.01
EndRepeat
```

Goto, Label

:Label

Goto Label

The **Goto** statement causes execution of a program to jump to the line that follows the **label** specified in the **Goto** statement. A **label** must begin with a **:** and can then be made up of any of the characters allowed in a variable name up to a maximum length of 30 characters.

Example

To loop without using the Repeat statement.

```
Set A = 1
:Start
Set A = A + 1
If A < 10 Then
    Wait For 0.01
    Goto Start
Endif
```

Routine, EndRoutine, Return, Call

Routine name

.

Return

.

EndRoutine

.

Call name [arg1 arg2 arg3 arg4 ...]

.

These statements allow callable **routines** to be implemented. All of the statements between the **Routine** and the **EndRoutine** form a subprogram that may be **called** from anywhere within the main program or another routine.

When a **Call** statement is executed, execution jumps to the first statement following the **Routine** statement that matches the name given in the **Call**.

Execution of a routine begins with the first statement following the **Routine** statement and continues until either a **Return** statement or an **EndRoutine** statement is reached.

Within a routine you can use a special variable called ARG to gain access to any parameters passed via the Call statement. Treat this variable like an array, so the first argument is ARG[0], the second ARG[1] and so on. If you SET an argument value (e.g. SET ARG[0] = 125) then the variable used as a parameter will take that value.

Example.

```

Program
  Call Subby "STRING1" 2.5 "STRING2" 3.5 COMBINED% TOTAL
EndProgram

```

```

Routine Subby
  Set S1% = ARG[0]
  Set S2% = ARG[2]
  Set V1 = ARG[1]
  Set V2 = ARG[3]
  * this will go into COMBINED%
  Set ARG[4] = S1% + S2%
  * this will go into TOTAL%
  Set ARG[5] = V1 + V2
EndRoutine

```

Example.

```

Set A = 0
Set B = 0
Repeat Until A > 100
  * Increment A by 3 and B by 4
  Call Increment 3 4
EndRepeat
.
.
.
Routine Increment
Set A = A + ARG[0]
If B >= 50 Then
  Return
Else
  Set B = B + ARG[1]
EndIf
EndRoutine

```

There is a special macro called **ARGCOUNT** that can be used within routines and which returns the number of arguments passed into the routine. This is very useful, especially when paired with the attribute **ISNUMERIC** which is implemented specifically with arguments in mind. **ISNUMERIC** returns **TRUE** (1) if the argument is of numeric type and **FALSE** (0) if it is of string type. Together they allow sophisticated routines to be constructed.

Example.

```

Program
  Set A = 1
  Set B = 2
  Set C = 3

  Call SUM A B TOTAL
  \\ TOTAL is 3
  \\
  Call SUM A B C TOTAL
  \\ TOTAL is 6

  Call BUILDSTRING "A" 3 "B" 2 "C" STRING%
  \\ Result is "ABBBCC" - arguments specified
  \\ "A" followed by 3 * "B" followed by 2 * "C"
EndProgram

```



```

Routine SUM
\\ Sums the first n-1 arguments and places the result in the final
argument.
  Set NARG = 0
  Set RET = 0

  Repeat (ARGCOUNT - 1)
    Set RET = RET + ARG[NARG]
    Set NARG = NARG + 1
    Wait For 0.001
  EndRepeat

  Set ARG[NARG] = RET
EndRoutine

Routine BUILDSTRING
\\ Concatenates the string arguments a number of times decided by the
numeric arguments.
  Set NARG = 0
  Set RET% = NULL%
  Set NREPEAT = 1
  Repeat (ARGCOUNT - 1)
    If ARG[NARG].ISNUMERIC = TRUE Then
      Set NREPEAT = ARG[NARG]
    Else
      Repeat (NREPEAT)
        Set RET% = RET% + ARG[NARG]
        Wait For 0.001
      EndRepeat
      Set NREPEAT = 1
    EndIf
    Set NARG = NARG + 1
    Wait For 0.001
  EndRepeat
  Set ARG[NARG] = RET%
EndRoutine

```

Notes.

- Recursion (both direct and indirect) is allowed but is not encouraged.
- There is no type checking on parameter passing when creating routines. Type checking is done at run-time and any errors reported via Spy. Similarly, there is no checking that the number of variables referenced by use of the ARG keyword matches the number of parameters passed in. Take care in using parameters and check Spy for errors when testing SLANG programs.
- Local variables used within a program are available to the routine. Regard all routines as being a part of the current program.
- A routine in one program cannot be called from another program.

Global Routines

A special class of routine exists known as a **Global Routine**. Global Routines are like any other routine except that they can be called from any program, not just the program within which they are defined. Global Routines are created by clicking the **New Global Routine** button on the toolbar.



This creates a new file that is in most respects identical to a program, it can even have routines of its own defined within it. The main difference between a Global Routine and a local Routine is in how they treat local variables. Routines within Programs can share the local variables of that program whereas Global Routines cannot. All variables used within a Global Routine are *private* to that routine (though, like programs, they can share them with local Routines defined within that Global Routine). This is best demonstrated by example.

Example.

```
Program
  Set A = 5

  Call INCREMENT
  \\ A now has the value 6

  Call ^GLOBAL1
  \\ A still has the value 6. The variable called A in ^GLOBAL1 is private
  \\ to that Global Routine and any local Routines within it
EndProgram

Routine INCREMENT
  Set A = A + 1
EndRoutine

\\ This is in a separate Global Routine file.
Routine ^GLOBAL1
  Set A = 20
  Call INCREMENT
  \\ A now has value 21, but this value is private to this Global Routine
  \\ and is not seen by any code outside of this file.
EndRoutine

Routine INCREMENT
  Set A = A + 1
EndRoutine
```

A major use of Global Routines is to enable you to create programs that can control many different, but largely identical, lines without having to duplicate code or use convoluted loops. For example, suppose we have three almost identical Processes; we can write one Global Routine that in effect replaces a Program and then write three very short Programs that simply call this Global Routine with a parameter that identifies the process, as shown in the example below.

Example.

```

Routine ^PROCESS
  Set NPROCESS = ARG[0]
  \\ Make a two-digit string variable of the process number
  Run REALTOSTRING2 NPROCESS 2 0 STRPROCESS%

  \\ Set up all of the indirect variables to be used
  Set &AI01.NAME = STRPROCESS% + "AI01"
  Set &AI02.NAME = STRPROCESS% + "AI02"
  Set &AI03.NAME = STRPROCESS% + "AI03"
  Set &AI04.NAME = STRPROCESS% + "AI04"
  Set &AO01.NAME = STRPROCESS% + "AO01"
  .
  .
  .

  \\ Begin the process
  Repeat Until &AI01 > 200
    If &AI02 > &AI03 Then
      Set VARVAL = VARVAL + &AI04
    .
    .
    .

    If NPROCESS <> 2 Then
  \\ This block is process specific
      Set VARVAL = VARVAL * $AI27
    EndIf
  EndIf
  .
  .
  .
  .
  Wait For 0.001
  EndRepeat
  .
  .
  .
EndRoutine

```

This is based on a system which has variables named 01AI01, 01AI02, 01AI03 etc. for process 1 and 02AI01, 02AI02, 02AI03 etc. for process 2 and so on. The first thing that the routine does is set up its indirect variables to map to the correct set of signals. The rest of the routine operates just using these indirect variables. It can of course also use normal signal notation (\$AI27) for variables that are not specifically process related.

The Global Routine would then be invoked from a program (as shown below) and a separate program created for each invocation required, passing in the correct parameter for the process number. This enables multiple copies of the routine – which is in essence our program – to run simultaneously. The fact that all local variables within a Global Routine are *private* means that there is a separate copy of them for each running Program, which is essential for the correct functioning of the Global Routines.

```

Program
  Call ^PROCESS 1
EndProgram

```

```

Program
  Call ^PROCESS 2
EndProgram

```

```

Program
  Call ^PROCESS 3
EndProgram

```

Making changes to a Global Routine will flag every program that uses that routine as having been modified, when looking at the SLANG Control Panel.

Include Files

Include Files provide another mechanism by which common code can be shared amongst many programs. To create an Include File click the New Include File button from the toolbar.



Include Files can contain any code that is found in a normal Program file.

```

IncludeFile INC1
  Set A = 1
  Set B = 2
  Set C = 3
  Set D = 4
  Set E = 5
  Set F = 6
EndIncludeFile

```

```

Program
  Include INC1
  Set SUM = A + B + C + D + E + F
EndProgram

```

```

Program
  Include INC1
  Set SUM = A * B * C / D / E / F
EndProgram

```

This example shows a set of initialisation statements that perhaps are required by several programs. Making a change to that include file automatically means the same change occurs in each file that includes it.

So why not just use a Global Routine to do this initialisation instead? Because as previously mentioned, local variables in a Global Routine are *private*, so the calling program would not see them and the initialisation would not have the required effect.

Routines can also be placed in Include Files but if they are then **ONLY** Routines should be in this file – do not intersperse it with normal code statements.

```

IncludeFile ROUTINELIBRARY
  Routine ROUTINE1
    Set A = 1
  EndRoutine
  \
  Routine ROUTINE2
    Set A = A * 2
  EndRoutine
  \
  Routine ROUTINE3
    Set B = A * 10
  EndRoutine
EndIncludeFile

```

Include Files that contain libraries of routines should be inserted at the bottom of Program files, after the **EndProgram** statement (i.e. in the place where Routine statements normally go). Having inserted an Include File full of Routines, those Routines are made available to the **Call** statement, as if they were actually written directly into the code.

```
Program
  Call ROUTINE1
  Call ROUTINE2
  Call ROUTINE3
EndProgram
Include ROUTINELIBRARY
```

Stop, Start

Stop name

Start name

These two statements allow other programs to be stopped or started by the current program. **Name** must be the name of another program that the SLANG runtime has successfully compiled (see the section on the SLANG runtime, later).

When a **Start** statement is executed the SLANG runtime is informed that the named program should be started. If this program is already running then it is left as it is, it is not **restarted**.

When a **Stop** statement is executed the SLANG runtime is informed that the named program should be stopped. It will be stopped **immediately** with no regard as to what it is currently doing. When writing SLANG programs that use this feature, be sure that **stopping** a program will not leave plant in a **dangerous** or **unknown** state!

Run

Run function [parameter [parameter]...]

This statement runs an external function from the SLANG Function Library. Depending on what the function is, there may be up to 10 parameters following the function name. Each of these parameters must be a constant or a variable (either string or numeric), again depending on the specific function. Note, only variables or constants can be passed, **not** expressions. Evaluate any expressions into a variable first and then pass the variable to the function.

The complete list of functions available is given in the appendix.

Comment

Inserts a comment into the program. The comment can be anything you like, and is ignored by the SLANG runtime.

Inline

This turns an existing statement into an inline comment. It can be used to temporarily remove a statement (say while debugging). Clicking the **Inline** button toggles the inline status of the current line.

Pause

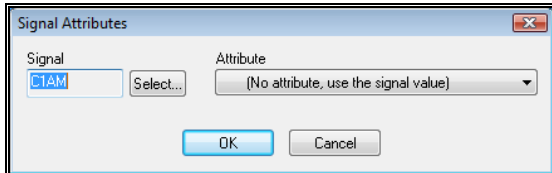
This causes the program to go into the paused state when this statement is executed. This has the same effect as pressing the pause button on the SLANG control panel (except that you are more in control of where the pause occurs with this method). It is used to place the program into pause mode so that you can single-step debug the program. You can use **Inline** to comment out these statements when debugging is finished, or you can delete them.

Signal Attributes

Some special attributes of signals are available to be read and written to. This list of attributes will grow as successive versions of Prodigy emerge, but the mechanism for accessing them will always be as described here, and no attributes will be removed.

There are two methods to access the attribute you need, one is to simply type a dot "." followed by the name of the attribute after the signal name. So for example \$TEMPFURN.UPDATEPERIOD would access the update period attribute of the signal TEMPFURN.

The second method is to choose the attribute from the Signal Attribute dialog. This is always displayed (when appropriate) after you have selected a signal to use in SLANG by using SigScan. This dialog is shown below.



The attributes that you can choose from are shown in the drop down list at the right of the dialog.

Expressions

Expressions may be **numeric** or **string**.

Numeric expressions are mathematical equations consisting of numeric constants, numeric variables and arithmetic operators. The operators that are allowed are the standard set: +, -, *, /. As is standard practice, * and / take precedence over + and -. In addition, the unary - is allowed when placed directly before a variable or constant with no following space, and this has the highest precedence..

Examples of correct and incorrect numeric expressions.

(Assume $A = 2$, $B = 3$ and $C = 4$).

$A + B * C$	<i>valid, equivalent to $A + (B * C)$ and equates to 14.</i>
$(A + B) * C$	<i>valid, equates to 20.</i>
$A * -B$	<i>valid, equates to -6.</i>
$A * - B$	<i>invalid, there should be no space between the - and the B.</i>
$A * D\%$	<i>invalid, mixing numeric and string variables (D%) is not allowed.</i>

String expressions consist only of string variables, string constants and the + operator.

Examples of correct and incorrect string expressions.

(Assume $A\% = \text{"The cat "}$ and $B\% = \text{"jumped over"}$).

$A\% + B\%$	<i>valid, equates to "The cat jumped over"</i>
$A\% + B\% + \text{" the dog"}$	<i>valid, equates to "The cat jumped over the dog"</i>
$A\% * B\%$	<i>invalid, only the + operator is allowed</i>
$A\% + D$	<i>invalid, mixing numeric and string variables is not allowed.</i>

Conditions

Conditions consist of one or more relational operations joined by **and** or **or** operators. As with arithmetic **expressions** the **and** operator takes precedence over the **or** operator, and in addition there is a unary **not** operator which may be prepended to relative operations.

RelationalOperation1 or RelationalOperation2 and RelationalOperation3

These relational operations are themselves formed by having two **expressions** separated by a relational operator.

Expression1 < Expression2

Expression1 <= Expression2

Expression1 > Expression2

Expression1 >= Expression2

Expression1 <> Expression2

Note the simplest form of an **expression** is a single variable or constant.

Examples of correct and incorrect conditions

Assume A = 2, B = 3, C = 4 and D% = "Cat"

A > B

valid, equates to false

Not A > B

invalid, Not is being applied to a variable and not a relational operation.

Not(A > B)

valid, equates to true

A + B >= C

valid, equates to true

A > B or C > B

valid, equates to true

C > B or A > B and C > A

valid, equates to false

(C > B or A > B) and C > A

valid, equates to true

A > B or D% = "Cat"

valid, equates to true

A > "Cat" or D% = "Cat"

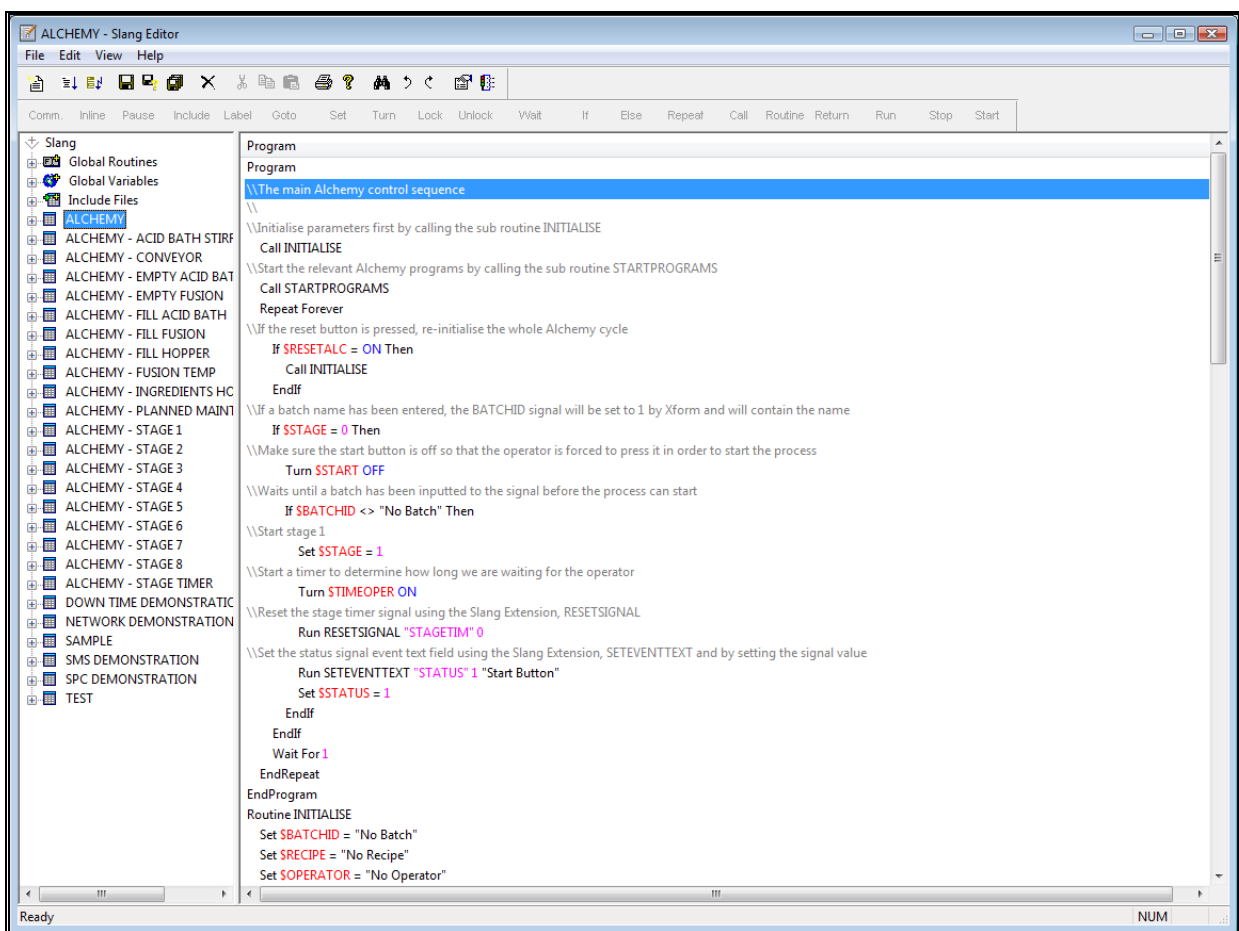
invalid, cannot mix numeric and string variables in a relational operation.

The SLANG Editor

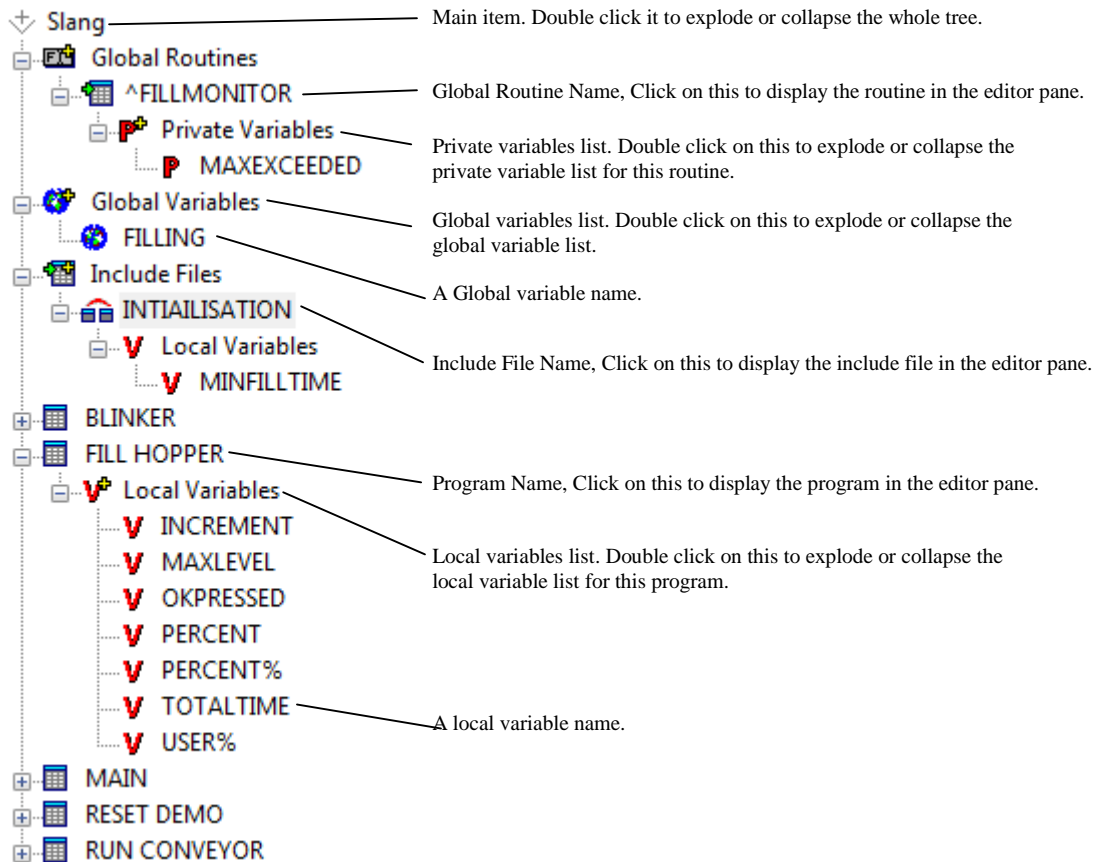
SLANG programs can be written in any text editor. When they are loaded into the *SLANG* Editor or the SLANG Runtime they are parsed to check their syntax. If any syntax errors are encountered a message is displayed showing the error that occurred, and loading of that program is abandoned. All errors must be eliminated before the program can be loaded.

The SLANG editor has been designed so that programs cannot be created with an incorrect syntax. It is fully context sensitive and will only allow the creation and removal of appropriate statements, depending on the statements surrounding the insertion point. Any variables and expressions entered via the editor are checked for validity as they are entered.

The SLANG editor is a multi-document editing environment which allows the user to switch rapidly from one program to the next whilst creating a suite of SLANG programs. The editor shows a list of all programs, together with lists of all the local variables used within each program and a list of all the global variables used throughout the suite of programs.



The left-hand pane of the editor is where the list of programs and variables are displayed. This area uses icons in a tree control to graphically represent the different aspects.



Here is an explanation of the items in the left-hand (selection) pane of the editor.

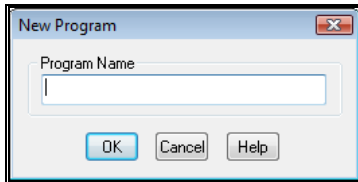
- **SLANG.**
This is the top-level item and serves only as a placeholder for the other items in the list. Double clicking on this item will collapse or explode the tree below it, which contains all of the programs that exist in the SLANG environment.
- **Global Variables.**
This item is a placeholder for the list of all global variables used throughout the programs in the SLANG environment. Double clicking on this item explodes or collapses the list of global variables.
- **Program Name.**
There is one program entry for each program that exists in the SLANG environment. Double clicking on an entry will explode or collapse the tree below this item, which contains an entry for the local variables used by the program. Clicking on a program name also has the effect of displaying the statements from the program in the right-hand (editing) pane.
- **Local Variables.**
This item is a placeholder for all of the local variables used by the currently selected program. Double clicking on this item explodes or collapses the list of local variables.

Creating A New Program

To create a new program click on the *New* icon or select *File/New...* from the menu.

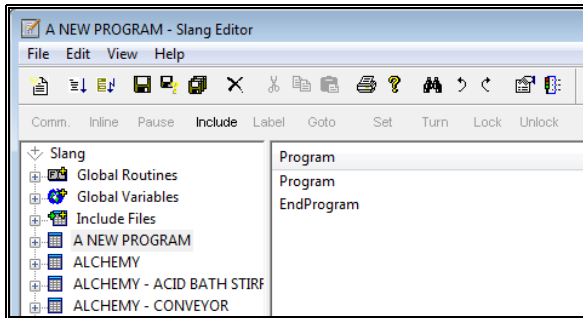


A dialog will then be displayed that requests the name of the Program that is to be created.



If a name is given that is already in use, an error message is displayed.

When a valid name has been given, an empty program unit will be created and the program will be appended to the end of the file list. Click on this entry to display the program in the editing pane. It will consist of two statements, **Program** and **EndProgram**.



You are now ready to add functionality to the program.

Adding Program Elements (Statements)

The statements that can be added to the program are defined in the Statement Toolbar that is positioned at the top of the editing window.



Reading from left to right these items are:

- **Comm.**
Comment. A string used to annotate the program but which does not actually form part of the executable program.
- **Inline.**
Turns a statement into an inline comment (has the effect of preventing the statement from being executed by the SLANG runtime).
- **Pause.**
Causes execution of the program to enter “pause” mode when this statement is reached. Useful only when debugging.
- **Label.**
A label which can be referred to by a Goto.
- **Goto.**
A statement which causes execution to jump to a given label, somewhere else in the program.
- **Set.**
A statement that assigns values to variables.
- **Turn.**
A statement that assigns 0 or 1 to a variable (like Set but geared more towards digital signals).
- **Lock.**
A statement that locks a variable, using the variable as a semaphore.
- **Unlock.**
Unlocks a variable locked by the statement described above.
- **Wait.**
A statement that will cause the program to pause until a certain condition is met or until a given amount of time has elapsed.

- **If.**
A statement that allows conditional execution. The lines following the If are only executed if a given condition is met.
- **Else.**
Optionally used with the If statement to allow for code that is only executed if the given condition is **not** met.
- **Repeat.**
A statement that repeats the following lines of code until a given condition is met or for a finite number of times, or forever.
- **Call.**
A statement that calls a subroutine within the current program.
- **Routine.**
Defines a subroutine within the current program.
- **Return.**
A statement that causes execution of a subroutine to stop. Execution will resume at the line of code following the Call statement that led to this subroutine being executed.
- **Run.**
A statement that executes an external subroutine that forms part of the Prodigy SLANG library.
- **Stop.**
A statement that causes a named program unit to stop executing.
- **Start.**
A statement that causes a named program unit to start executing.

In the picture of the toolbar shown above, all of the items on it are shown as being available. In use, this is never the case. Only those items that can legally be added at the current insertion point will be available.

The current insertion point is defined as being the line of code in the editing pane that is currently selected. When adding statements they are added **below** the current insertion point and the insertion point is moved on so that the new statement becomes the current insertion point. In this way, successive statements can be added without having to continually move the insertion point manually.

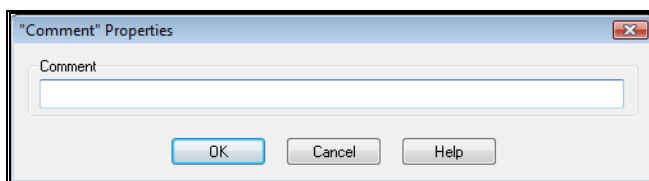
When inserting statements from the **Paste** buffer (these having been placed there by an earlier **Copy** or **Cut** operation), the statements are inserted **above** the current insertion point.

When you click on an item in the Statements toolbar the first thing that happens is that a dialog is displayed requesting whatever information is required to complete that statement. These dialogs are specific to the type of statement being added and are described in detail below. If you decide not to add the statement after all, simply click the **Cancel** button on the dialog rather than the **OK**. When you click the **OK** button, any fields you have filled in on the dialog are validated and if there are any errors, a dialog is displayed showing the line of code that would be generated by the dialog, along with a message indicating what the error is. Correct the error and try pressing **OK** again.

Comment



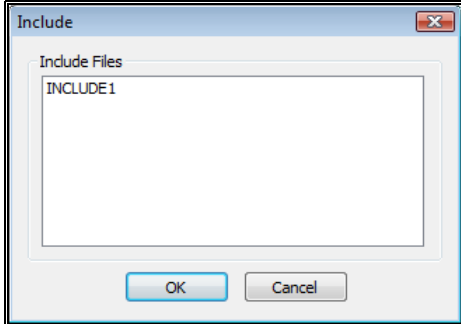
Enter any text that you want to appear as a comment. Entering nothing and just pressing **OK** will add an empty comment field to the program, which can be useful for spacing out lines of the code to aid readability. No validation is carried out on the comment.



Include



Used to add an include file into a piece of SLANG code. If the include file contains a set of routines it **MUST** be placed at the end of the program. If not it can be included anywhere within the SLANG program.



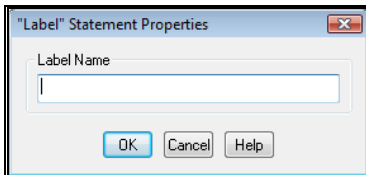
A statement will be generated as follows.

Include IncludeFileName

Label



A label is used to mark a point in the program that you will want execution to jump to by using a Goto statement somewhere else in the program. Use this dialog to enter the name of the label. The label name given must not already exist and its maximum length is 12 characters.



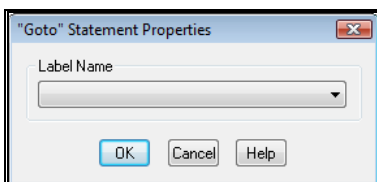
A statement will be generated as follows.

:label

Goto



Used to cause program execution to jump to another point in the program, this point being the label specified (see Label, above). Select the required label from the drop down list.



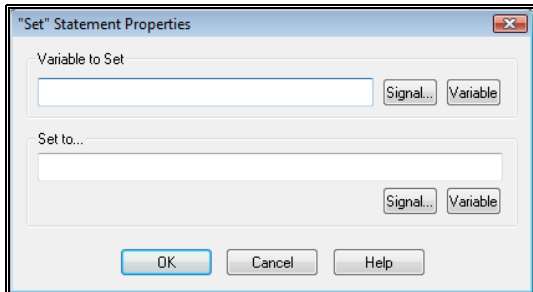
A statement will be generated as follows.

Goto label

Set



The Set statement is used to assign a value to a variable. The value is taken from an expression that can simply be another variable, a constant or an arithmetic equation. This dialog lets you enter the variable and expression to be used.



- **Variable To Set.**

Enter the name of the variable you wish to set. There are three scopes of variable, **Global**, **Local** and **Signal** and two **types** of variable, **Numeric** and **String** as well as **arrays**.

 - A Global variable may be accessed by every program within the SLANG environment and is represented by having a ^ as its first character. Global variables have a maximum length of 30 characters.
 - A Signal variable is actually a reference to a signal within Prodigy's signal database. Assigning a value to a Signal variable has the effect of changing the value of that signal. Any programs within Prodigy that are interested in that signal (e.g. Display Builder, Alarm Display) will be informed of its new value and will act accordingly. A signal variable is represented by having a \$ as its first character and has a maximum length of 8 characters. By definition, Signal variables are global.
 - A Local variable is accessible only by the program in which it is used. Another program may have a local variable of the same name, but it **not** the same variable. Local variables consist solely of the characters that form its name and have a maximum length of 30 characters.
 - All variables are **numeric** unless they have a % as their final character. If this is the case the variable is defined as being **string**. The % character is treated as being part of the variable name so it is permissible to have a numeric variable called ABC and a string variable called ABC%. They will be treated entirely differently and are not interchangeable. You cannot have a variable that is a Signal and a String (i.e. its name begins with a \$ and ends with a %).
 - The **Signal...** button may be used to select a signal that is to be set. The standard SigScan dialog is displayed when this button is pressed.
 - The **Variable...** button may be pressed to display the variable selection dialog (see below).
- **Set To...**

Enter an expression that is to be evaluated to provide a value that the variable will be set to. The simplest form of an expression is a numeric constant or another variable. More complicated expressions consist of a mixture of variables and constants, along with arithmetic operators such as + or *.

 - Pressing the **Signal...** button allows a signal to be chosen from the SigScan dialog and its name will be inserted at the current cursor position. The **Variable...** button allows a variable to be selected. If an existing variable is highlighted first, it will be replaced by the new signal or variable selection.
 - See the section on the SLANG language earlier in this manual for more details on expressions. The expression may be a numeric or a string expression, and again this is described in the language section.

A multitude of validation takes place when **OK** is pressed.

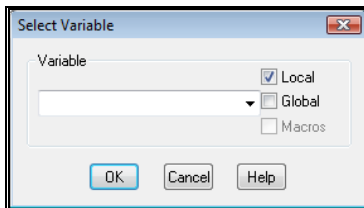
- The variable name must be of the correct length for the type of variable that it represents.
- The expression is parsed to ensure that the correct syntax has been entered.
- The type of the expression (string or numeric) is checked against the type of the variable.

- Any variables that are referenced but do not already exist result in a dialog being displayed that requests confirmation that the variable is to be created. This is to minimise the likelihood of variable names being mistyped. If confirmation is not given the variable will not be created and the statement will not be added. You can then correct the variable name and try again.
- If a Signal variable is used that does not actually exist in the signal database, a warning is issued. This is because the SLANG editor cannot add signals to the signal database. You must do this via the **Signal Database Configuration** program.

If all validation is passed a statement of the form

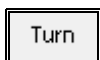
Set *variable* = *expression* is added to the edit window.

Variable Selection

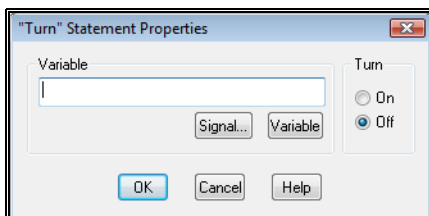


This dialog contains a drop down list of the variables currently in use in the current SLANG program. This list can contain any or all of local **variables**, **global variables** or **macros**, depending on what is selected from the check boxes on the dialog (macros are only available for the **Set to** field). You can select the required variable from the list, or type in the name of a new one.

Turn



The Turn statement sets a variable to 1 or 0. Its functionality can be achieved by using the Set statement, but the Turn statement's dialog is geared towards digital signals.



- **Variable.**
Type in the name of the variable (which must be numeric) or signal required. Alternatively Signals and variables can be selected by pressing the **Signal...** or **Variable...** button (see the section on the **Set** statement for more details). The SigScan dialog that is displayed will show only Digital signals.
- **Turn.**
On or **Off.** Sets the specified variable to 1 or 0 respectively.

Validation is minimal and is limited to checking that the variable is of the correct type and length and prompting if it does not yet exist.

One of the following statements will be generated.

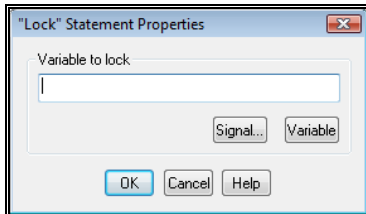
Turn variable On

Turn variable Off

Lock



The Lock statement is used to turn a variable (of any type, and of either global class) into a semaphore. Semaphores are used to ensure **mutual exclusion** or **indivisibility** within the programs in the SLANG environment. This means that it provides a mechanism to prevent one program from proceeding with a course of action whilst another program is proceeding with its own particular course of action.



It works as follows:

- Program A attempts to **Lock** variable ^SEM1.
- The attempt succeeds. Program A is free to continue executing statements.
- Program B attempts to **Lock** variable ^SEM1.
- It fails, as ^SEM1 is already **Locked** by program A. Program B suspends execution until it is able to successfully **Lock** ^SEM1.
- Program A comes to the end of its mutually exclusive section and **Unlocks** ^SEM1.
- Program B can now **Lock** ^SEM1 and continue executing.

The following points are worthy of note.

- It does not necessarily have to be Program A that unlocks ^SEM1. Any program can unlock a variable that may have been locked by any other program.
- If a program attempts to lock a variable that it has already locked, it will still wait for that variable to be unlocked. It does not assume that "as I already have it locked, I am free to proceed". The reason for this is that, in conjunction with the previous point, the program may be waiting for another program to unlock the variable. You may, for instance, **Lock** a variable, **Start** another SLANG program and then try to **Lock** the variable again. The second program may perform some operations and then, as its final action, **Unlock** the variable, in effect telling the original program that it has finished.
- Locking a variable does not prevent its value being changed by another program. If that is the intention then all programs that may modify that variable should **lock** it, **set** it then **unlock** it. Note that if this mechanism is used with Signal variables, it will **not** prevent the signal value being altered by an external source, only by other SLANG programs.
- While any variable of global class can act as a semaphore, it makes most sense to use Global variables rather than Signal variables. This is because it is best practice to create variables whose sole use is that of a semaphore.

The Lock dialog allows you to enter the name of the variable that you wish to lock. The **Signal...** or **Variable...** buttons may be used to display the relevant selection dialogs button (see the section on the **Set** statement for more details).

A statement is generated as shown below.

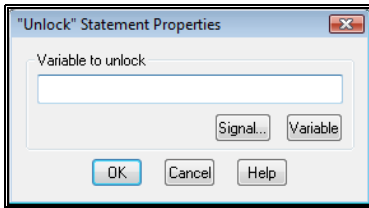
Lock variable

Unlock



This is used to unlock a variable that has been previously locked, either by this program or by another. The dialog allows you to enter the name of the variable that you wish to unlock. The **Signal...** or **Variable...**

buttons may be used to display the relevant selection dialog button (see the section on the **Set** statement for more details).



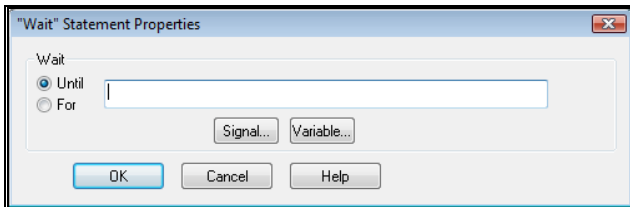
A statement is generated as shown below.

Unlock variable

Wait



The wait statement causes a program to pause until either a given condition has been satisfied or a given amount of time has elapsed.



- **Until.**
If this option is selected then you must enter a condition into the field. A condition is a logical statement that evaluates to either true or false. Conditions are explained in more detail in the section describing the SLANG language.
- **For.**
If this option is selected then you must enter an arithmetic expression which when evaluated will equate to the amount of seconds that the program will wait for. This number may be fractional (e.g. an value of 0.1 will cause the program to pause for 1/10th of a second). By implication, the expression must be numeric and not string. See the section on the SLANG language for more details on expressions.

The **Signal...** or **Variable...** buttons may be pressed to display the relevant selection dialog button (see the section on the **Set** statement for more details).

When the **OK** button is pressed the condition or expression is validated and any errors encountered will be displayed.

One of the statement types shown below will be generated.

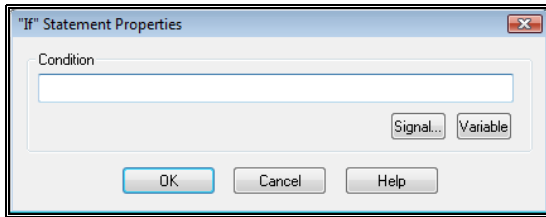
Wait Until condition

Wait For expression

If

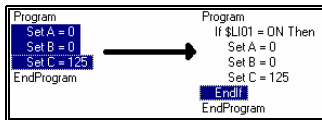


The If statement is used to provide a mechanism for code to be executed only if a given condition is true. Enter the condition that must be matched into the field on the dialog. The **Signal...** and **Variable...** buttons may be used to display the relevant selection dialogs button (see the section on the **Set** statement for more details).



When you insert an **If** statement the matching **Endif** is also inserted directly below it. This is because an If statement must always have a matching Endif and the SLANG editor will not allow you to enter anything that is not syntactically correct.

If you select a range of lines in the code rather than just one, the If and Endif will be inserted so that they **enclose** the range of lines.



Select a range of lines by clicking on the first line in the range and then shift-and-clicking on the last line in the range. The SLANG editor checks the range that has been selected and if it partially overlaps any **block** statements, the statement toolbar will be modified so that you cannot add any statements. If the selection is valid, the toolbar is modified so that only those statements that are allowed to be placed around a group selection (If-Endif and Repeat-EndRepeat) are available.

A **block** statement is any statement that must always exist as a matching pair. Such statements are **If-Endif**, **Repeat-EndRepeat**, **Routine-EndRoutine**, and **Program-EndProgram**

When the code is executed, if the condition is true the statements between the **If** and the **Endif** are executed. If the statement is false, execution jumps to the statement following the **Endif** (see also the **Else** statement, below).

The if statements will be added as shown below

If condition Then

.

Endif

Else



(There is no dialog for the **Else** statement, it is simply inserted into the code as soon as the statement button is pressed).

The Else statement is only available when the insertion point is within an **If-Endif** block. Further, it is only available if there is not already an **Else** statement in the block.

When the code in an **If-Else-Endif** block is executed, if the condition is true the statements between the **If** and the **Else** are executed. If the statement is false, the statements between the **Else** and the **Endif** are executed.

The statement will be added as shown below

If condition Then

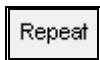
.

Else

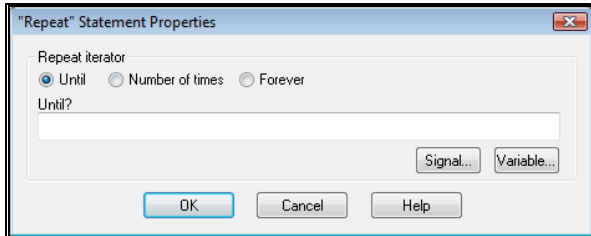
.

Endif

Repeat

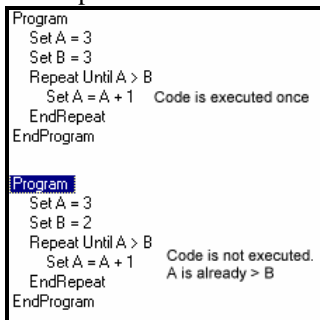


The repeat statement is used to provide a mechanism for repeating a block of code either for a given number of times or until a condition is satisfied.



- **Until.**

If this option is selected, enter a condition into the field on the dialog. When the code is executed, the statements between the **Repeat** and **EndRepeat** will be executed repeatedly until the condition is true. Note, if the condition is true when the **Repeat** statement is first executed, the code between the **Repeat** and **EndRepeat** will not be executed. In other words, the condition is evaluated at the **top** of the loop and not at the **bottom**.



- **Number of times.**

If this option is selected, enter an expression in the field on the dialog. When the code is executed the expression is evaluated and the code between the **Repeat** and **EndRepeat** is executed that number of times. (**Notes:** The code in the **Repeat-EndRepeat** block will only be executed if the expression yields a result greater than 0. The expression is only evaluate once - changing the values of any variables used in the expression will have no effect on the number of times the loop is executed. Only the integer part of the expression result can be used).

- **Forever.**

The repeat statement will execute forever. You may exit the repeat loop at some point with a **goto** or a **return** statement, or you may simply repeat the loop forever.

When you insert a **Repeat** statement a matching **EndRepeat** is also inserted. As with the **If-Endif** statements, a **Repeat-EndRepeat** block can be inserted around a group of statements. See the **If** statement for more details.

The **Signal...** and **Variable...** buttons may be used to display the relevant selection dialogs button (see the section on the **Set** statement for more details).

The statement that is inserted into the code will be:

Repeat Until *condition*

Or

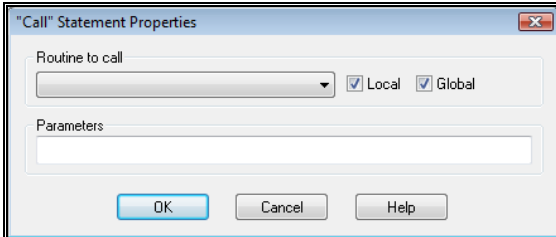
Repeat *expression*

Followed by **EndRepeat**.

Call



The **Call** statement is used in conjunction with **Global Routines** and the **Routine** statement to implement subroutines within SLANG. When you **Call** a **Routine** execution passes from the line of code containing the **Call** to the first line of code within the given **Routine**. When the **Routine** has completed, execution returns to the line of code following the **Call** statement.

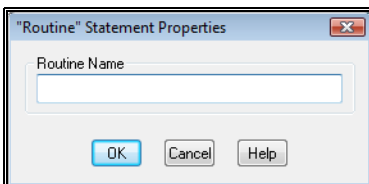


In the dialog shown above, you must select the name of the **Routine** you wish to **Call** from the list of those available. This mechanism therefore forces you to define your **Routine** before you can make a **Call** to it. Below the **Routine to Call** is the parameters section, **variables** can be passed by entering them into this section (described earlier in **Statements** section). You can filter on **Local** and **Global** routines using the appropriate check boxes.

Routine



The **Routine** statement is used to define a subroutine that can be called from a program or another **Routine**. Enter the name of the routine into the field on the dialog. If a routine with that name already exists and error will be generated.



When the **Routine** statement is entered a matching **EndRoutine** statement is inserted after it. The **Routine** button on statement toolbar is only available if the insertion point is on an **EndRoutine** or an **EndProgram** statement.

When a **Routine** is being executed, if the **EndRoutine** is reached execution returns to the point from which the **Call** to the **Routine** was made.

The statements appear as shown below.

Routine name

EndRoutine

Return



The **Return** statement is used to force execution of a **Routine** to stop and return to the point from which the **Call** to the **Routine** was made. The **Return** statement can only be added within a **Routine** block (i.e. it cannot be added within the **Program** block) and any number of **Returns** can be added. Here is an automatic implied **Return** associated with the **EndRoutine** statement.

There is no dialog for the **Return** statement, it is added straight into the code. It appears as shown below.

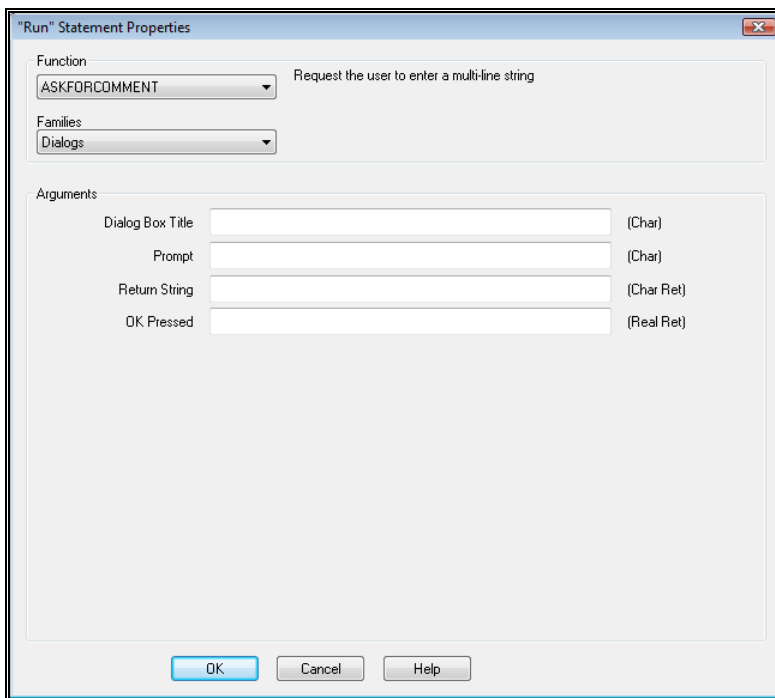
Return

Run



The run statement gives access to the SLANG extensions runtime library. Each entry in the library provides an element of functionality that could not be achieved by the SLANG language alone. Key areas of functionality provided include:

- User-interface functions (via dialog boxes).
- Alarm functions.
- Signal database manipulation
- Batch manipulation.



A **Run** statement consists of the keyword **Run** followed by the name of the function that is to be run. This is then followed by a number of parameters, the type and number of which will vary from function to function.

The **Run** statement dialog changes dynamically depending on which function is selected from the **Function** list. The contents of the **Function** list change depending on which family is selected from the **Families** list.

When a function is selected the parameter list for that function is displayed in the **Arguments** section of the dialog. The name of the argument is displayed to the left of the relevant field and the type of argument required is shown to the right of the field. A brief description of the function is given to the right of the **Function** list.

If the type of argument required is **Char** the argument must be a **string constant** or a **string variable**. If the type is **Real** the argument must be a **numeric constant** or a **numeric variable**.

If the word **Ref** appears after either **Char** or **Real** it means that the function will place a value back into that argument. If this is the case the argument **must** be a variable of the relevant type – a constant cannot be used.

All parameters must be filled in. To enter a blank string use the null string placeholder NULL%, do not just leave the field blank. String constants must be enclosed in quotes ("), if not the words of the string will be interpreted as variable names!

Expressions cannot be entered, only variables or constants. Evaluate any expressions into variables first (using the **Set** statement) and then pass the variable as a parameter to the function.

In the example shown above, the parameters would be filled in as follows.

Arguments		
Dialog Box Title	"Comment Request"	(Char)
Prompt	"Please enter a comment"	(Char)
Return String	Ret%	(Char Ret)
OK Pressed	OkPressed	(Real Ret)

As an extra aid a tooltip appears when you hover over each field giving more information about the field.

Dialog Box Title	"Comment Request"	(Char)
Prompt	"Please enter a comment"	(Char)

The message that will be displayed on the top bar of the pop-up dialog

The statement will appear in the code as follows.

Run ASKFORCOMMENT "Comment Required" "Please enter a comment" RET% OKPRESSED

When the statement is executed the following dialog will appear.

The user will be required to enter a string. If they press **OK** the string will be placed into the (string) variable **RET%** and a value of 1 will be placed in the variable **OKPRESSED**. If **Cancel** is pressed, a value of 0 is placed into **OKPRESSED** and nothing is done with **RET%**. This means that the SLANG program can test the value of **OKPRESSED** to determine whether or not **OK** was pressed on the dialog.

Note that while the dialog is visible the SLANG program that invoked it is in a paused state. Execution of that program resumes when either **OK** or **Cancel** is pressed. This is true for all **Run** statements that invoke a dialog.

A full list of SLANG extension library functions is given in the appendix.

Stop

Stop

The **Stop** statement allows one SLANG program to terminate any other SLANG program, including its self.

Select the name of the required program from the list, or type in the name of a program if the program does not yet exist. If the program does **not** exist, a warning will be issued. If you leave the entry blank, SLANG assumes you mean **Stop** the current program.

When this statement is executed, the named program will be stopped immediately, regardless of what it is currently doing. This could leave plant in a dangerous or unknown state, so it is up to you, the programmer, to ensure that this does not happen.

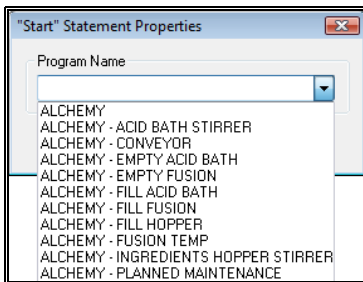
The statement appears in the code as

Stop name

Start



In the same way that the **Stop** statement allows any SLANG program to stop any other SLANG program, the **Start** statement allows any SLANG program to start any other program running.



Select the name of the required program from the list, or type in the name of a program if the program does not yet exist. If the program does **not** exist, a warning will be issued.

When this statement is executed, the named program will be started immediately, unless it is already running. If it **is** already running, the statement has no effect. It **does not** restart the program – in order to achieve this you must **Stop** and then **Start** the program.

The statement appears in the code as

Start name

Modifying A Program

Selecting Programs

Select the program you wish to modify by clicking on it in the selection pane (the left-hand pane of the SLANG editor). The program will be displayed in the editing pane (the right-hand pane).

You can work on more than one program at once. Switch between programs by clicking on the required program in the selection pane and then working on them in the editing pane.

Saving Programs

When a program has been modified an asterisk (*) is displayed to the right of its name in the selection pane. To save the changes made to a program click on the **Save** icon or select **Save** from the **File** menu. This will save the currently selected program.

To save programs under a different name select the **Save As** icon or **Save As** from the **File** menu.

To save all programs that have been modified click on the **Save All** icon or select **Save All** from the **File** menu. On the image below the icons are from left to right **Save**, **Save As** and **Save All** respectively.



Deleting Programs



To delete an entire program, click on it in the selection pane and press the **Delete** or **Del** key or click the **Delete** icon. The editor will prompt to see if you really want to delete the program. If you do, reply **Yes**. Be aware that when a program is deleted, it is removed from the SLANG runtime environment **immediately**. If the program is running, this could have serious consequences.

Exiting From The Editor

Exit the editor by pressing the **x** box in the top right corner of the editor window, or by selecting **Exit** from the **File** menu. If any programs have been modified and not saved you will be asked, on a program-by-program basis, whether or not you want to save the changes. Alternatively, you will have the option to **Cancel** the exit procedure, which will result in you remaining in the SLANG editor.

Abandoning Changes

If you have made modifications to a program but do not want to save them, **Quit** the SLANG editor. Do this by pressing the **x** box in the top right corner of the editor window, or select **Exit** from the **File** menu. When prompted if you wish to save your changes, reply **No**.

Adding Statements

The procedure for adding new statements has already been described in the earlier section on creating programs.

Modifying Statements

To modify a statement simply double click on it in the editing pane. The relevant dialog will be displayed to allow you to alter the properties of the statement. The rules for what can be entered in the dialogs are exactly the same as when adding statements (see the earlier section). Clicking on **Cancel** within a properties dialog will have the effect of leaving the statement unchanged.

Deleting Statements

To delete a statement, click on the required statement and press the **Delete** or **Del** key. The statement will be deleted and the statement below the statement that was deleted will become the currently selected statement.

Some statements are known as **block** statements, which means that they exist as a matching pair of statements that enclose other statements. When looking at the editor pane, such statements are highlighted by having the statements that they enclose *indented* a level. Block statements within SLANG are:

- **If-Else-Endif**
- **Repeat-EndRepeat**
- **Routine-EndRoutine**

When you delete a **Repeat** or **If** statement you will be asked if you want to delete the whole block, including the statements within it, or just the block statement its self. For instance, if you delete the **Repeat** statement in the example below, you would have the choice of deleting all 6 statements or just the **Repeat** and **EndRepeat** statements.

```
Repeat Until (A > 200)
  Set A = A + 1
  Set $AI01 = $AI01 + A
  Set B = $AI01 * 2
  Call FILLER
EndRepeat
```

The deletion of a block statement is treated the same whether you try to delete the **Repeat** or the **EndRepeat** statement (or the corresponding part of the **If** or **Routine** block statement).

In addition, if you have selected the **Else** statement in an **If-Else-Endif** block, you will have the choice of just deleting the **Else**, deleting all statements in the **If-Else-Endif** block or deleting the **If**, **Else** and **Endif** statements.

When you delete a **Routine** (or **EndRoutine**) statement the only option available is to delete **all** statements in the block. A warning will be issued asking if this is what you really want to do.

Multiple Deletion

You can select multiple statements (by using the shift-click technique) and delete all of the statements at once. This is only allowed, though, if no block statements are included in the selection. This is because it would be too complicated and confusing a process to go through the procedure mentioned above for deleting block statements when there may be several such statements in the selection.

Copying & Pasting

Single or multiple lines can be selected for copying. Unlike multiple deletion, **Copying** is allowed when there are block statements in the selection. The proviso here is that the **entire** block must be within the selection. If a selection starts or ends mid way through a block statement, copying will not be allowed.

When you have made your selection either click on the **Copy** icon, select **Copy** from the **Edit** menu or press **ctrl-C**. The items that you have selected are copied to the SLANG editors clipboard (**note** they cannot be pasted into other applications).

To **Paste** the clipboard into the program (any SLANG program may be selected – you can copy from one and paste into another) select the statement **above which** you want the clipboard to be inserted. Now either click on the **Paste** icon, select **Paste** from the **Edit** menu or press **ctrl-V**. The SLANG editor will paste the contents of the clipboard into the program, above the statement you currently have selected.

After pasting, the contents of the clipboard remain unchanged and can be re-pasted elsewhere if required.



Cutting & Pasting

The procedure for cutting and pasting is exactly the same as for copying and pasting except that when you **Cut** the statements, they are removed from the program. To **Cut** a statement or statements, click on the **Cut** icon, select **Cut** from the **Edit** menu or press **ctrl-X**.

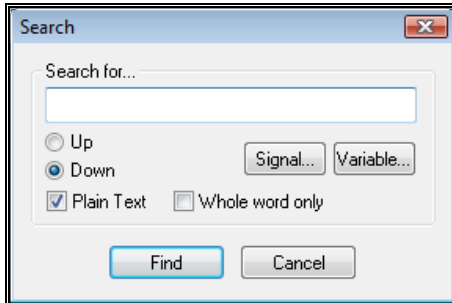


The Cut Icon

Find



Click the **Find** icon to define what you want to search for.



- **Search For.**
Enter the string to be searched for – this can be a signal, variable or any text. If you specify a signal, enter the \$ sign too.
- **Up/Down.**
Specifies the direction in which to search, based on the current statement as the starting point.
- **Plain Text.**
If this box is checked each statement of each program will be searched to see if it contains the relevant text, irrespective of whether it is a variable, a statement name or even a parameter in a **Run** statement.
- **Whole word only.**
If this is checked then the system must find the string specified as a complete word, and not part of some other word. So if we search for **dog** as plain text, whole word only it would be found in **set dog = 1** but not in **set dogger = 1**.

*If **Plain Text** is not checked, the system searches only for variables that match, and **whole word only** is irrelevant.*

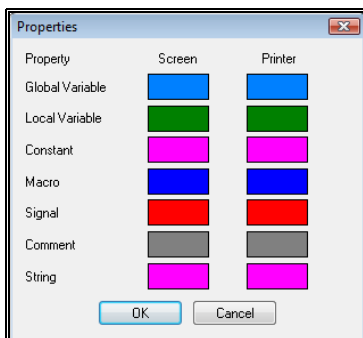
The system looks from the first program in the list to the last looking for the selection. When it finds a match it displays the line of the program that matches. If it passes beyond the end (or start if searching up) of a program, it moves on to the next one. If it goes past the end (or start) of the list of programs, it continues at the other end.

The **Find Next** and **Find Previous** buttons find the next or previous occurrence of the selected search.

Display Properties



The SLANG editor shows certain parts of the program in different colours (variables, constants, etc.). Clicking the Display Properties icon (shown above) displays the following dialog.



Click on any colours that you want to modify and select new values for them from the standard colour picker. Note that you can choose different colours for printout than are used on the screen if you so wish.

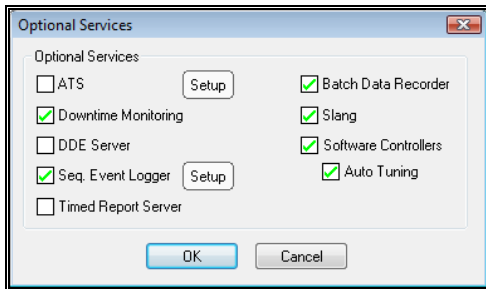
SLANG Control Panel



The final Icon (shown on the right above) is the **SLANG Control Panel** icon. Clicking this will launch the SLANG control panel.

The SLANG Runtime Environment

The SLANG runtime environment is an optional item on the **Prodigy Control Panel**.



If the item is checked, the SLANG runtime environment will be started when the services start up. The runtime performs the following actions:

- The SLANG runtime scans the SLANG directory (**\$/Configuration/SLANG**) for all programs that have been created. Each program that it finds is compiled and any errors that are encountered are displayed in a dialog.
- If there are compilation errors in a program, that program is flagged as such within the runtime, and it will not be allowed (in fact it will not be able) to run. Any other programs (which may have already compiled successfully) but which attempt to **Stop** or **Start** a program which has failed to compile, will also be flagged as having compilation errors.
- Any programs which are marked as **AutoStart** (see below) will begin executing. All others will be marked as Stopped.

While the runtime is active it performs the following actions regarding programs that are modified in some way:

- If a new SLANG program is created, it is automatically loaded into the runtime, but its **AutoRun** status will be **false**, so it will not begin executing.
- If a program is deleted it will be removed from the runtime **immediately**. Be aware of this when deleting programs.
- As the user edits SLANG programs, they are not reloaded into the runtime automatically. The user must manually force a reload by using the SLANG Control Panel. This is because:
 - The user may not want the new program to be used yet.
 - The program may not be complete or contain bugs.
 - The program may be currently running and to reload it would have undesirable effects.

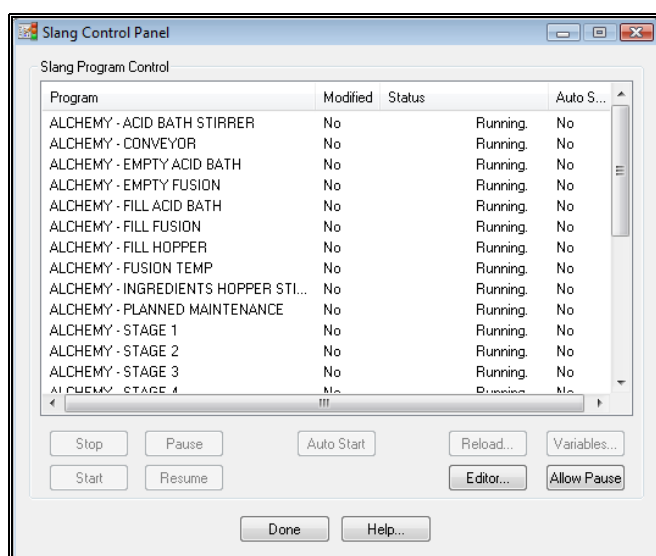
The user can interact with the SLANG runtime via the **SLANG Control Panel**. This will allow the user to Stop, Start, Pause, Step, Resume and Reload SLANG programs. The SLANG Control Panel is also used to define which programs should be automatically started when the runtime environment is started.

The SLANG Control Panel

The SLANG Control Panel allows users to interact with the SLANG runtime environment. Specific actions that can be performed are:

- Stop a program that is currently running.
- Start a program that is currently stopped.
- Pause a program that is currently running.
- Single-step a paused program.
- Resume a program that is currently paused.
- Reload a program that has been modified.
- Set programs to be AutoStart or not.
- View the current values of variables and signals used in the SLANG programs.

The SLANG Control Panel is shown below.



The dialog shows a list of all of the programs that exist within the SLANG runtime, together with their status and whether or not they have been **Modified** and are **AutoStart**. The status that a program can have will be one of the following.

- **Stopped.**
The program is not running. When it begins running, it will start with the first line of code in the program.
- **Running.**
The program is currently running.
- **Paused.**
The program has been paused (from the SLANG Control Panel). It is currently not executing, but when execution resumes it will be from the line of code at which it is currently stopped.
- **Compile Error.**
There was an error when the program was compiled. The program cannot be started until the errors have been fixed. As it is almost impossible to produce a program that will not compile when using the SLANG editor, one of the following causes is most likely to blame.
 - The program was produced or edited outside of the SLANG editor.
 - The program references signals that do not exist in the signal database. This condition is detected by the SLANG editor but results in a warning rather than a fatal error.
 - The program references (via a **Start** or a **Stop** statement) a program that either does not exist or did not compile. Using **Start** or **Stop** with a program that is not compiled is detected by the SLANG editor and results in a warning. If one program does not compile, it can cause a ripple

effect as any program that references it will be flagged as not compiled and any program that in turn references this program will also be flagged. And so on...

- **Program not read.**
The SLANG runtime was unable to open the file. The reason for this is indeterminate and you should check the file via the explorer.
- **Unknown.**
The SLANG Control Panel is requesting the current status of each program from the runtime environment and is awaiting a response.

Additionally, if a program has been modified (via the **SLANG Editor**, for instance) the **Modified** column of the list will indicate this. If this flag shows **true**, the program that resides on disk is not the same one as is loaded into the SLANG runtime environment. Programs are not automatically reloaded when they are modified, in case the program is not complete. The program must be reloaded by using the **Reload** button on the control panel.

Be aware that modified programs will be loaded when **Prodigy** is next restarted, so do not leave modifications half completed when working on an active program.

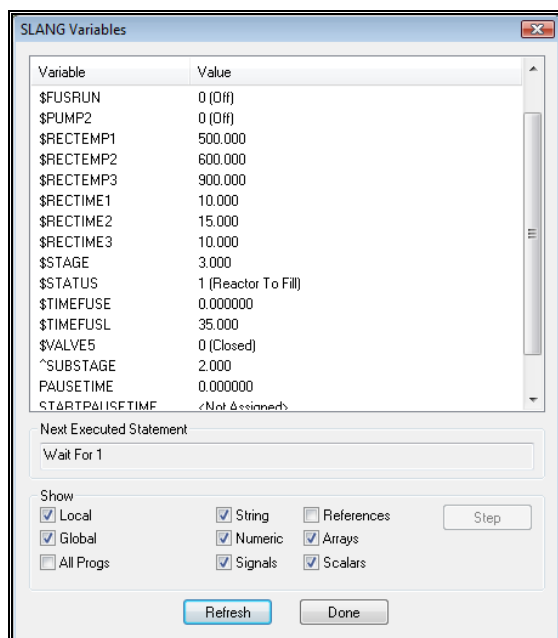
Select a program from the list by clicking on it. The buttons below the list then allow a range of actions to be performed on the program. The actions of these buttons are described below.

- **Stop.**
This button is only available if the status of the program is **Running** or **Paused**. It stops the program from executing any further, regardless of what it is currently doing.
- **Start.**
This button is only available if the status of the program is **Stopped**. It begins execution of a program starting at the first line of code.
- **Pause.**
This button is used to temporarily suspend a program. It is only available if the status is **Running**. The runtime will not execute any more statements from this program until it is resumed.
- **Resume.**
This button is only available if the status is **Paused** and has the effect of causing execution to continue again, from the point at which it was paused.
- **AutoStart.**
Toggles the programs AutoStart state (shown in the list alongside its Status). If AutoStart is Yes, the program will automatically begin executing when the SLANG runtime is started.
- **Reload.**
This button is only available if the status is **Stopped**. It causes the runtime to reload the program to incorporate any changes that may have been made to it. If it compiles OK its status will be **Stopped**, otherwise it will be **Compile Error**.
- **Editor.**
This launches the SLANG Editor program.
- **Variables.**
This displays a dialog showing the current value of variables used within SLANG. You can also launch that dialog by double-clicking on a program from the list.
- **Allow Pause.**
This allows the **Pause** statement to have an effect (i.e. to actually Pause a program). If you have Pause statements in your programs they will have no effect until this button is pressed.

The Variables Dialog

When a program is paused, either from the control panel or with a Pause statement within the program, you can then single step the program in order to debug it. You can also view the state of the variables (local, global and signals) used within SLANG – in fact this latter action can be done even when the program is not paused.

The **Variables** dialog is as follows.



The parts of this dialog are as follows.

- **Variable List.**
This is a list of all of the variables that match the criteria selected below, along with their values. If the variable is an array, the maximum array bound that has been accessed is shown, and you must double click on the row to display a further dialog which contains the values of each array element.
- **Last Statement Executed.**
When the program is running ordinarily this is the statement that was executing when the dialog was displayed, or when **Refresh** was clicked. When single-stepping it is the statement that is waiting to be executed.
- **Local.**
If this box is checked, local variables are shown in the variables list.
- **Global.**
If this box is checked, global variables are shown in the variables list.
- **String.**
If this box is checked all string variables are shown.
- **Numeric.**
If this box is checked all numeric variables are shown.
- **Signals.**
If this box is checked all signals used within SLANG programs are shown.
- **References.**
If this box is checked all referenced signal variables are shown.
- **Arrays.**
If this box is checked all array type variables are shown.
- **Scalars.**
If this box is checked all scalar variables (i.e. variables that are not arrays) are shown.
- **All Progs.**
If this box is selected then the variables list will show the values of all signals and global variables

used throughout all SLANG programs. If it is not checked only those variables and signals that are used in the currently selected program are shown.

- **Step.**
Pressing this button executes the statement currently showing in **Last Statement Executed** and progresses to the next statement to be executed. The **Last Statement Executed** field is updated, as are any variables that are being displayed.

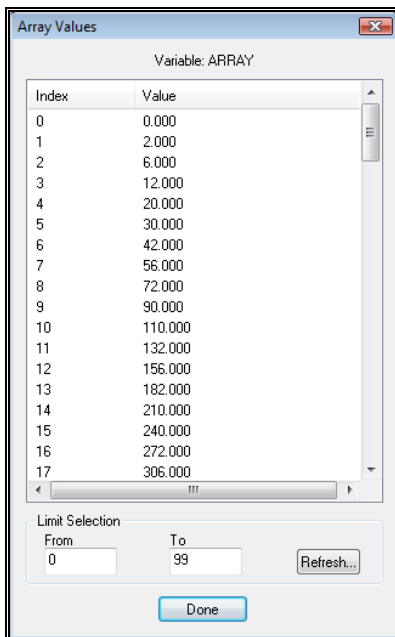
Note that the variable type selections work in combination, so you can select for example “all global variables that are arrays of strings and that are only used in this program”.

When arrays are used the display shows the array with its maximum accessed index (in this example, 99)

Variable	Value
ARRAY	[99]
INDEX	118.000
OKPRESSED	1.000
RET%	Warning Tripped
TEMP	236.000

You can double-click on the line showing the array to get a list of all elements in the array (see below).

The Array Variable Dialog



This displays a list of the values of all of the elements of the array. Initially only the first 100 elements are shown, this being specified by the **From** and **To** fields (if there are 100 or less elements in the array, the **To** field reflects this number). You can alter the **From** and **To** values to display different sections of the array, or indeed to display it all no matter how large the array is.

For really large arrays this could have a real performance effect.

If our SLANG program specifies the maximum array size as 100

set ARRAY1[] = 100

and we try to access elements beyond that array size (so elements above 100) an error message is issued to **Spy**. If we are trying to set the illegal array element

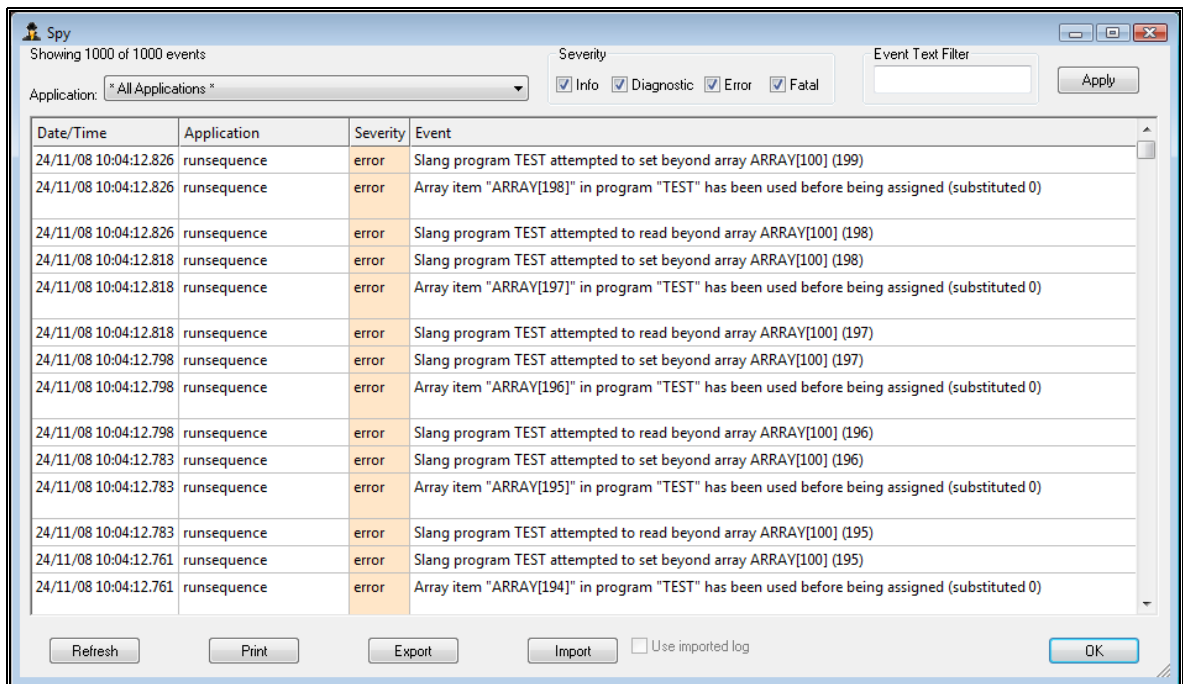
set ARRAY1[150] = 549

an error message is issued but no other action is performed. If however we were trying to read the illegal array element

set ijk = ARRAY1[145]

an error message is issued to **Spy** and the value 0 (or an empty string if the array is an array of strings) is used in place of the illegal array element.

And example of **Spy** output from a program that sets an array limit of 100 but tries uses the array within an iterative loop that goes up to 200 is shown below:



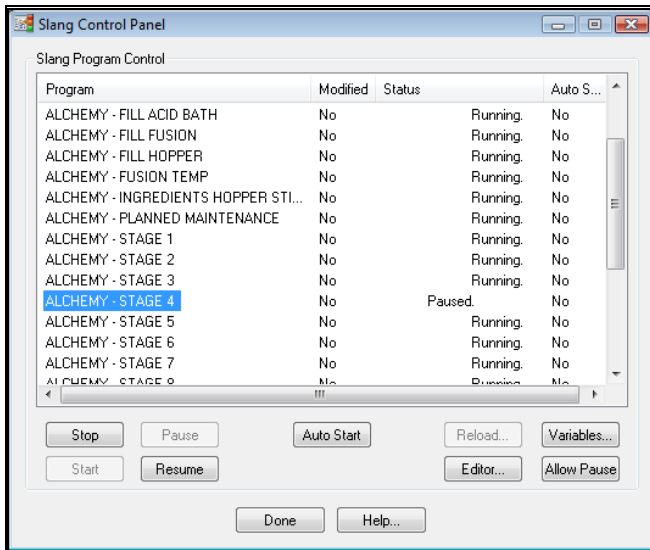
Date/Time	Application	Severity	Event
24/11/08 10:04:12.826	runsequence	error	Slang program TEST attempted to set beyond array ARRAY[100] (199)
24/11/08 10:04:12.826	runsequence	error	Array item "ARRAY[198]" in program "TEST" has been used before being assigned (substituted 0)
24/11/08 10:04:12.826	runsequence	error	Slang program TEST attempted to read beyond array ARRAY[100] (198)
24/11/08 10:04:12.818	runsequence	error	Slang program TEST attempted to set beyond array ARRAY[100] (198)
24/11/08 10:04:12.818	runsequence	error	Array item "ARRAY[197]" in program "TEST" has been used before being assigned (substituted 0)
24/11/08 10:04:12.818	runsequence	error	Slang program TEST attempted to read beyond array ARRAY[100] (197)
24/11/08 10:04:12.798	runsequence	error	Slang program TEST attempted to set beyond array ARRAY[100] (197)
24/11/08 10:04:12.798	runsequence	error	Array item "ARRAY[196]" in program "TEST" has been used before being assigned (substituted 0)
24/11/08 10:04:12.798	runsequence	error	Slang program TEST attempted to read beyond array ARRAY[100] (196)
24/11/08 10:04:12.783	runsequence	error	Slang program TEST attempted to set beyond array ARRAY[100] (196)
24/11/08 10:04:12.783	runsequence	error	Array item "ARRAY[195]" in program "TEST" has been used before being assigned (substituted 0)
24/11/08 10:04:12.783	runsequence	error	Slang program TEST attempted to read beyond array ARRAY[100] (195)
24/11/08 10:04:12.761	runsequence	error	Slang program TEST attempted to set beyond array ARRAY[100] (195)
24/11/08 10:04:12.761	runsequence	error	Array item "ARRAY[194]" in program "TEST" has been used before being assigned (substituted 0)

Debugging SLANG Programs

The Variables dialog is the key element of debugging SLANG programs. Not only does it allow you to monitor the state of variables throughout the SLANG system but it also allows you to single step programs. There are two ways to enter single-stepping mode of a program.

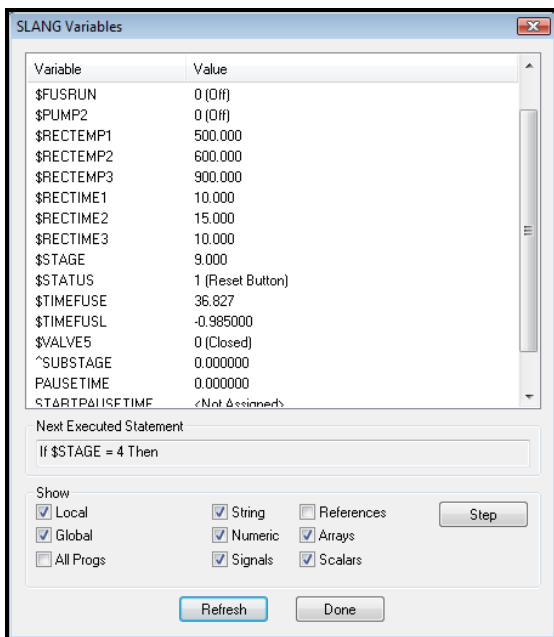
The first method is to put a **PAUSE** statement into a SLANG program. When the program executes and it reaches this statement it will pause, which is to say that it will stop at the line with the pause on it and wait for you to either single-step or **resume** it.

Note that you must have pressed the Allow Pause button on the SLANG Control Panel for the program in question in order for the PAUSE statement to have an effect. This is so those programs with PAUSE statements can execute normally and not pause unless you are actively debugging it.



The second method is just to press the **Pause** button on the SLANG Control Panel (above). This method is less useful as you cannot control where the pause will occur (usually it will be at a **WAIT** statement as this is the place where a SLANG program will spend most of its idle time).

You single-step SLANG programs that have been paused (by either method) by bringing up the **Variables** dialog and clicking the **Step** button (see below).



This will execute the statement currently shown in **Next Executed Statement** and any changes to variables that this causes will be reflected in the variables window when **Step** is pressed.

It is no simple matter to debug a series of concurrent and interlinked SLANG programs – try to keep your designs as simple as possible.

Introduction to OPC

OPC stands for OLE (Object-Linking and Embedding) for Process Control. It has been established by 'The OPC Foundation' to provide a standard mechanism for communicating with numerous data sources such as hardware devices or databases. The OPC Specification and related material form a set of standard OLE/COM interface protocols based on the Microsoft's OLE/COM technology.

This technology defines standard objects, methods and properties that are used by software applications to communicate information.

From the point of view of getting raw data from physical devices into a SCADA system such as Prodigy, the aim of OPC is to reduce having to write a different driver for every device it is required to talk to. A driver is still required to talk to the hardware at the lowest level obviously (an OPC server), but it communicates the data using a predefined set of interfaces as defined by the OPC Foundation. This data can be read, again using the OPC standard interfaces and passed on to any interested software by writing an OPC Client. OPC Servers are currently available for most hardware, either from the manufacturer or a third party. As the interfaces for communication between servers and clients are pre determined, only one client is needed as an interface between multiple servers and a SCADA package such as Prodigy.

At the moment OPC is still being developed and new OPC interfaces are still currently being introduced such as alarm and event handling, however it is up to the individual servers how much of the OPC standard they implement. From Prodigy's point of view we need to get the raw measurements in and send values out we do not need to use some of the newer more complicated interfaces. Using Prodigy's OPC Client, which behaves similarly to the other drivers within Prodigy in that it is started from the Driver Configuration, in theory any device with an available OPC server can be communicated to with the minimum of effort. Prodigy's OPC Driver (Client) will connect to multiple OPC Servers.

OPC Servers will probably need some sort of configuration as to which values are to be read or written. This will vary considerably between servers and may involve often getting familiar with new software!

Servers Groups and Items

Servers are the actual drivers that communicate with the hardware or possibly software in some cases. A Server registers itself when it is installed specifying its name and the path where it is to be found. All servers have unique names and are started and stopped when necessary by OPC Clients that connect to it. Several clients may communicate with a server simultaneously.

A Client will communicate with a server by referring to groups and items. Items are the actual signals or values that the server knows about. Each OPC Server Item has a unique name. Items are put into groups according to their update rate (in milliseconds) and deadband (a percentage change after which notification is given from the server to the client). A Client will only be given the item values when they change beyond the deadband specified. The server does not pass on values on the update rate to the client but will inform the client if it cannot achieve the requested update rate.

Each server has its own names for its items, which are unique within that server. It tells the client about its items if is requested to do so. Some servers when being configured put part of the item's name under the user control. Others have their own names that cannot be changed. Sometimes the item's name is made up of meaningful text separated by a '.'. For Example:

COM2.2500 Unit.IO.Module02.M02_C1.MeasV

Or PROFibus_1.Master3.001_Slave.Tag02

The OPC Software

Prodigy's OPC Software suite consists of the following:

- An **OPC Client** or **driver** which behaves as other drivers, except instead of talking directly to the hardware modules it talks to any number of **OPC Servers**. Each OPC Server will communicate directly with the hardware.

- **Prodigy OPC Address Dialog**, which maps signals within Prodigy to, items within OPC Servers. The Address Dialog has been designed so that once it has been set up intelligently it can be used efficiently to configure signals without the need to continually interrogate servers.
- An **OPC Server** that can pass signals from within Prodigy out to other sources using **OPC Clients**.
- An **OPC Server Configuration** program that tells the Prodigy OPC Server which signals within Prodigy we are interested in sending elsewhere.
- An **OPC Client** that is independent of Prodigy and can be used when Prodigy is either running or not. This is useful as a diagnostic tool to see the items in any installed server and their various properties.
- A **Test OPC Server**, which again is useful as a diagnostic tool or as an aid in finding how OPC works. This server has some simulated items continually changing.

Using OPC to Communicate between Prodigy and Hardware Modules

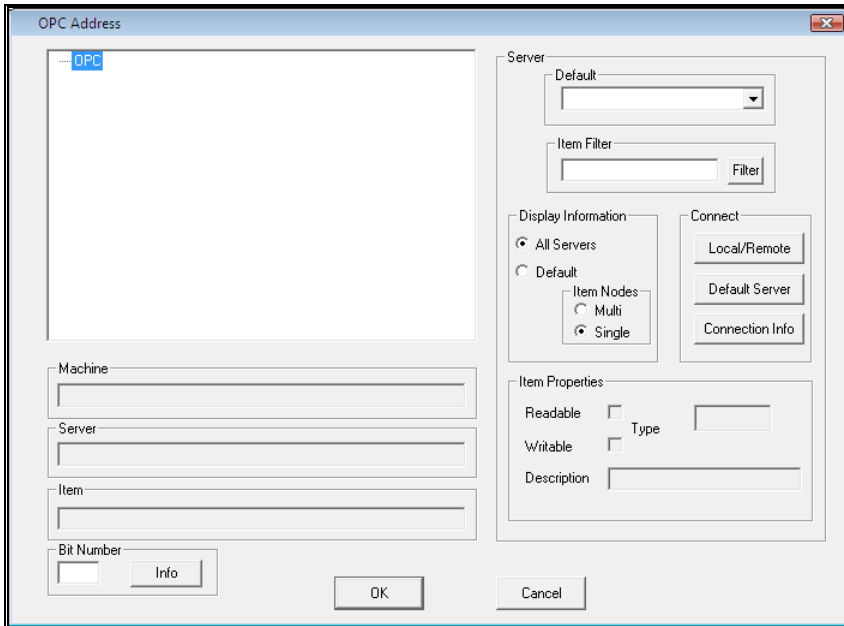
- An OPC Server needs to be available which communicates with the hardware. These are available from a variety of sources such as the hardware manufacturers or other software manufacturers. The server needs to be installed on the PC connected to the hardware devices and some sort of configuration exercise is likely. The nature of this will vary between servers.
- Once the server has been installed and configured the independent client (**OPCClient.exe**) may be used to check that the items are all available from the server and the OPC communications are ok. This step is not essential but is useful for verifying that everything is satisfactory so far.
- Next use the **Driver Configuration** to put the **Prodigy OPC Driver** into one of the driver slots.
- Create some signals in the **Signal Database**.
 - Specify the attributes required such as type.
 - Specify the driver for all these signals to be **Prodigy OPC Driver**.
 - Then right click on **Hardware Address** to bring up the address Dialog. This initially looks more complicated than most of the other Address Dialog windows. However by carefully selecting a default server and filtering, the process of mapping Prodigy signals onto OPC Items should be straightforward. See the “**Mapping Prodigy Signals onto OPC Items**” section for a more detailed explanation of how to use the OPC Address Dialog.
- Press the “Go Live” button so that the newly created signals will become live in the signal database.
- Put the OPC Signals on a **Display Builder** Display (or use **Signal Monitor**) to check that they are receiving input. Assuming everything has been configured correctly and that the required OPC Servers and the Prodigy OPC Driver is running, the values should now be coming in via the OPC Server and going into the Prodigy system via the Prodigy OPC Driver.

Mapping Prodigy signals onto OPC Items

OPC Servers know about the values coming in and going out from the hardware they are talking to. These are called OPC Items. When a server is interrogated for the items it knows about; it may send only the items the user requires or it may also send many that they do not require. It is necessary to choose the required signals from the server and map them onto Prodigy signals in Prodigy’s signal database.

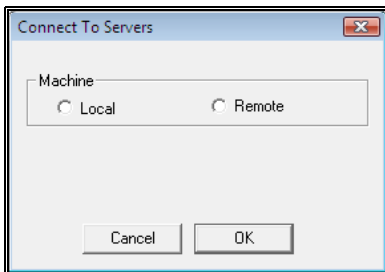
This is done using the **OPC Address Dialog**. It has the ability to display all the items on servers currently known about or a selection of items on one default server. It is time consuming to interrogate servers and this would be unacceptable and unnecessary when going in and out of the address dialog. Therefore the program keeps its own copy of the items last known about on the hard disk. This is much faster and saves having to connect and disconnect to the servers every time. At any time the user may reconnect to the servers and get a totally up to date list of items from them.

Usually, once you have a list of servers you would select the default server to be the one you are currently interested in mapping the items from. You would filter item names to make it easier to select the item you require. You would then click the Default radio button to display only those items from the default server, which match the criteria of the filter. This selection will be remembered by the address dialog on exit and will be the one displayed when you re enter the OPC address dialog. The address help dialog is shown below:



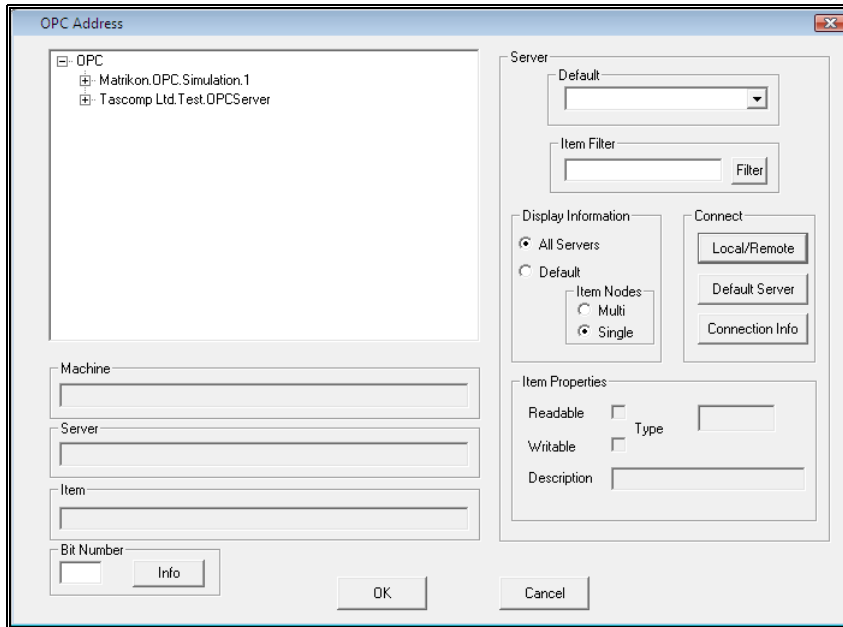
Connecting to OPC Servers

Initially when the dialog is entered for the first time, there will be no servers displayed. You will need to connect using the **Local/Remote** button. This will bring up the following dialog:



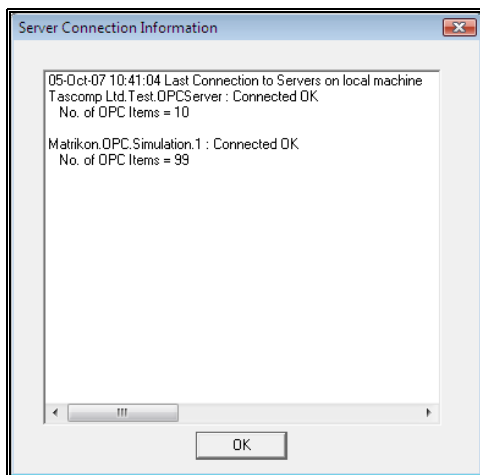
You may connect to all servers on either the local machine or a remote machine. If **Remote** is chosen you will be prompted for the name of the machine you wish to find the servers on. It is worth noting here that not all servers have the ability to communicate remotely. It is necessary to check this ability exists with any appropriate servers.

On acknowledging that this is really what you wish to do the program will search for all the installed servers on the machine and attempt to connect to them and find the names of the items they know about. This may take a short amount of time to do. A screen similar to the one below may then be displayed.



The server information is displayed using a tree structure. If the OPC Item names contain '.' these are used as delimiters to create the tree levels. This is standard practice.

Clicking the **Connection Info** button shows a record of the connections attempted.



This gives a list of all the servers that a connection was attempted on, and whether or not an error occurred. If no error occurred the number of OPC Items known about is displayed. Only the servers that were connected to without an error are displayed in the tree structure although a record of all attempted connections will be displayed in the **Server Connection Information** list. If you expect to see a server that isn't listed or one that returned an error it is possible that the server hasn't been installed properly or it hasn't registered itself.

Alternatively the connection to the Server was ok but the item required is not shown, it is likely the Server has not been configured properly and either doesn't know about the item or doesn't remember it on start-up. Possibly the Server configuration has not been saved. See the individual OPC Server and configuration documentation for more information.

Setting a Default Server

Once the server you are interested in is displayed along with its item names, it makes sense to select this as the default server.

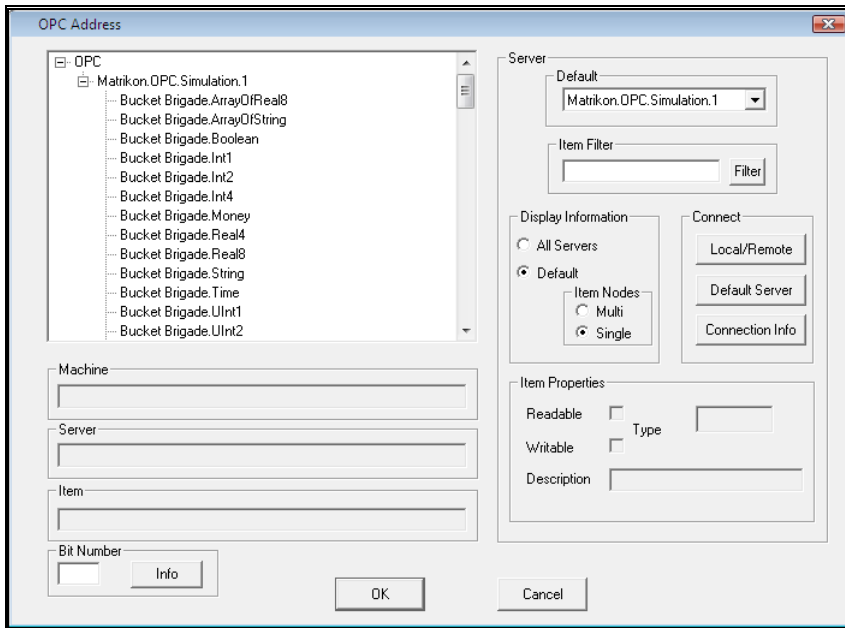
- Select the required server from the **Server** list in the **Default** section.
- Check the **Default** button in the **Display Server Information** section.

Only the selected server will now be displayed, which helps to clear up the display if there are multiple servers. You can select a different server simply by choosing it from the **Server** dropdown list.

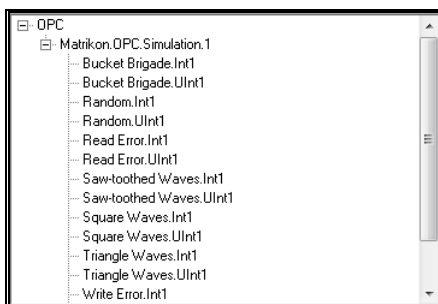
Filtering

Filtering will display a subset of the items available. Item names from the default server may be filtered on text strings. Enter a string into the **Item Filter** box and click the **Filter** button. The example below shows the OPC Address dialog before and after filtering. (Note: the filter text is case sensitive)

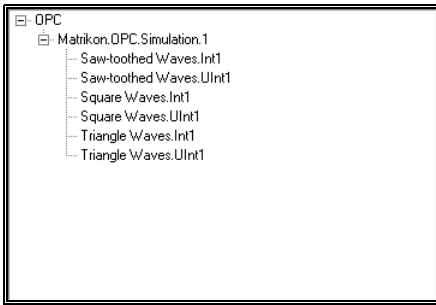
Before:



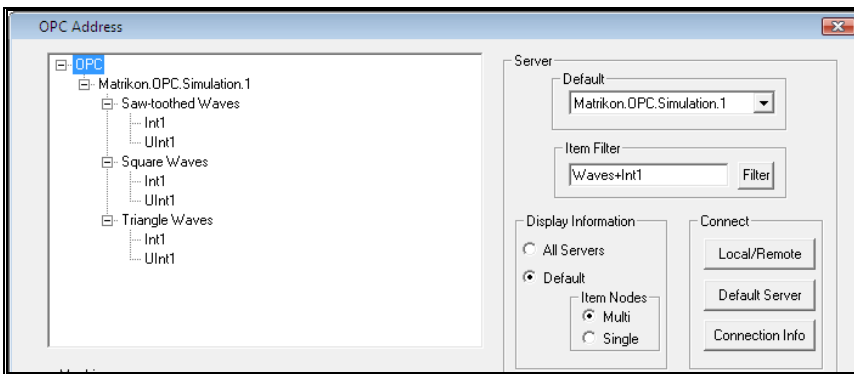
After, showing only those items that contain the string **Int1**:



You can also select items that contain multiple strings. The example below contains the strings **Waves** and **Int1** and is selected by the string **Waves+Int1**:



The default items may be displayed as a flat tree structure as shown in the examples above, or as a multi-tree structure as shown below. Choose which option you want by clicking the **Multi** or **Single** button.

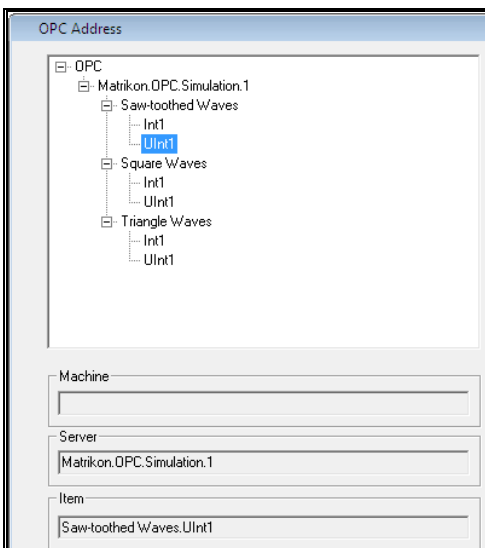


Selecting an OPC Item

The OPC item required is selected by double clicking on the appropriate item on the display.

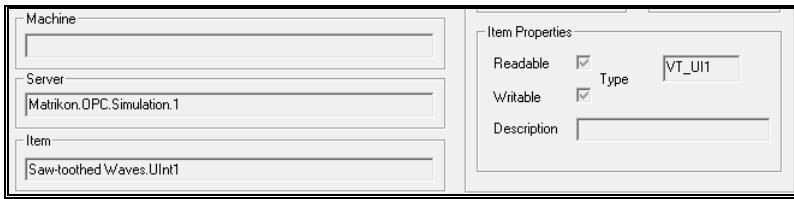
The display can be showing only the default server items in multi or in single format, or it can be showing all connected server items – it makes no difference.

The selected item will then shown underneath the main display.



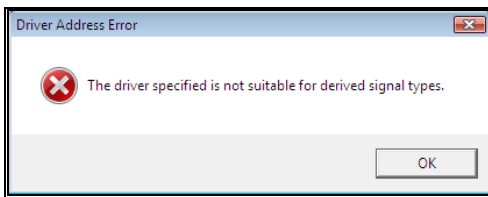
This is the item that will be mapped to your Prodigy signal on exit from this address dialog. If the local machine is being used the 'Machine' field will be blank.

Various properties of the selected item are also shown, and these are for reference only.

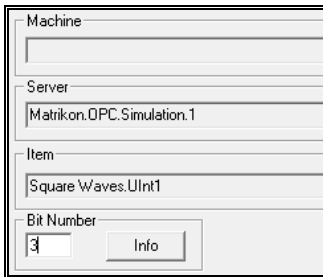


The **Readable**, **Writable** and **Type** attributes should always be shown but a description will only appear if the server has implemented this facility.

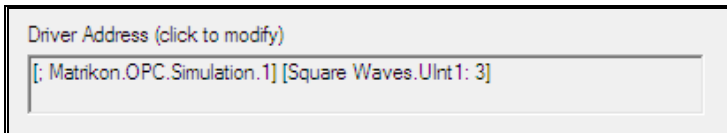
If there is a conflict between the Prodigy signal type and the OPC item type, the user will be informed either when trying to enter the OPC address dialog (if derived signals are used) or when they exits it, the dialog is shown below.



Sometimes, especially with digital signals, several values will be combined into one OPC Item but will be represented in Prodigy as multiple signals. In this instance specify the appropriate bit number.



On exit from the OPC address dialog the current information displayed is remembered so that the next time the address dialog is entered, the display is as it was when exited. The driver address on the Signal Database Configuration dialog will be filled in with information that identifies this OPC item (below).



The OPC Driver

Once the signals in Prodigy’s signal database have been mapped onto OPC items from an OPC Server the Prodigy OPC Driver may be started from the Driver configuration screen in a similar way to any other driver.

It is communicating with OPC servers which are software, so doesn’t need any serial port information inputting. Once up and running it receives information from the servers so this is shown in the Rx Count. It only receives information from Servers when values change, so the rate at which the Rx count is increasing by will vary depending on the change of values on the plant. (This has nothing to do with the Prodigy update date rate). There will not be any information in the Tx Count.

The driver connects to the servers required and organises the items into groups according to their update rate. It then receives updates of the items from the server as they change and schedules the values to and from prodigy as the other drivers do.

The OPC Driver is written as an OPC client interfacing between Prodigy and the OPC servers.

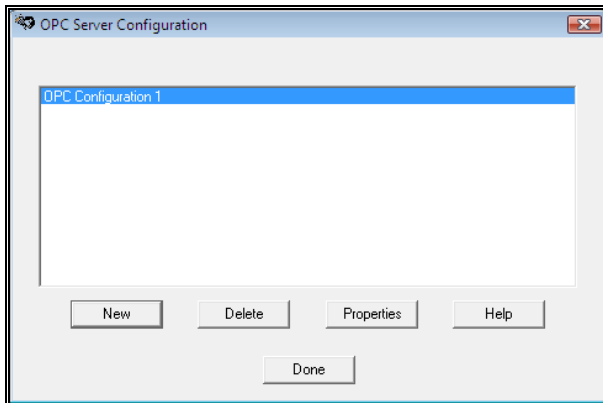
The OPC Server

It may be required to pass signal values from Prodigy to other sources. One method of doing this is to use the Prodigy OPC Server. The appropriate signals are configured using the OPC Server Configuration Program (See Later). One server looks after all the signals in any number of configuration files. The Server will pass on any changes of signal value from within Prodigy to interested clients within the limits specified in the configuration and also allow updates of writeable signals from clients to Prodigy.

The OPC Server is started automatically by any OPC client that requests signals from it.

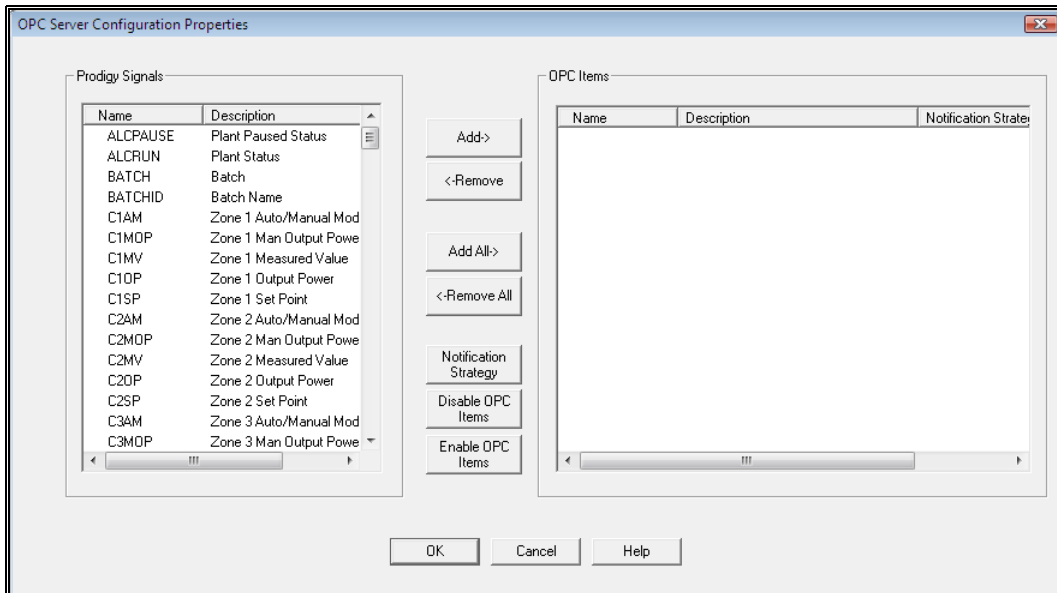
OPC Server Configuration

This is used to tell the Prodigy OPC server which Prodigy signals are of interest. The following screen is displayed.



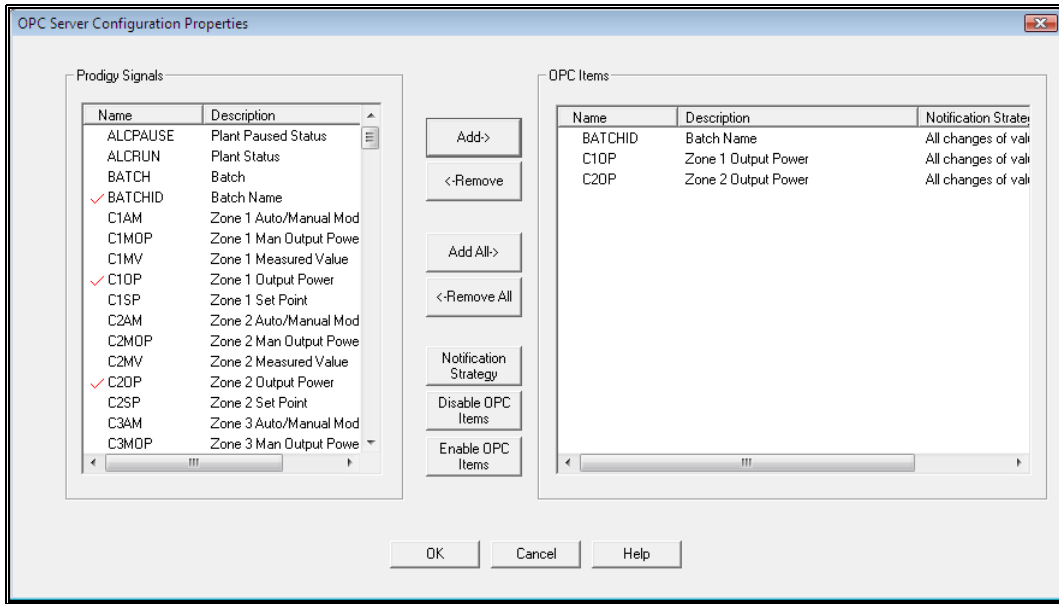
Create a new configuration by clicking the 'New' button and giving it a unique name. Several configurations of OPC items may be made. The OPC Server will read all configuration files to check which signals it should know about.

Click the properties button or double-click on the name to bring up the following screen.



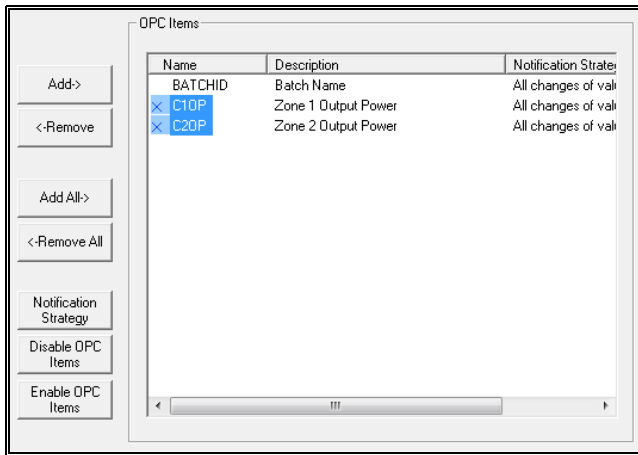
A list of all the available signals in Prodigy is displayed on the left-hand side. Adding a signal to the right-hand side will select it as being an OPC Item. The Prodigy OPC Server will then monitor it if it is running and it will be available to be accessed by any OPC Clients that connect to the server.

Signals may be added one at a time, all added or multiply selected. Signals that have been selected as OPC Signals appear on the right-hand side and are marked on the left-hand side by a tick.



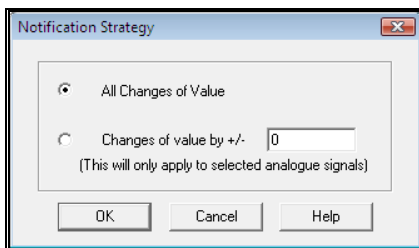
OPC Signals can be removed in the same way.

OPC Items can be disabled, which is a way of stopping them from being made available to interested clients without having to delete them from the configuration. This is more convenient if they are only to be removed temporarily. If the removal is deemed to be permanent then it is better to remove them from the configuration fully. Disable signals by selecting as many as are required and pressing the **Disable OPC Items** button.



Disabled items are marked with a cross, as shown above. You can re-enable them by selecting them and clicking the **Enable OPC Items** button.

Analogue signals may have a notification strategy applied to them. By default *all* changes of value of the signal are passed on to interested clients. Setting a Notification Strategy allows you to limit this by specifying that the signal *must change by a given amount* before anything is passed on to interested clients.



If the same signal occurs in more than one configuration file the most frequent notification strategy is used.

Diagnostic Tools

In order to check that the OPC connections are working without needing to have Prodigy running there are two diagnostic tools provided. These may run with or without Prodigy running.

- The OPC client may be used to check that values are coming from the OPC servers. In theory if the OPC Items can be seen by this program, which has a user-friendly interface they can be seen by the Prodigy OPC driver. If this program cannot read or write the OPC Items expected from the servers, it is possible that the fault lies in the configuration of the server.
- This standalone OPC Client communicates with servers and allows the user to set up groups and items and dynamically updates a display of their values. Configurations of various servers and items may be saved and loaded in at a later date. Output values may also be sent to devices.

Using the Standalone OPC Client

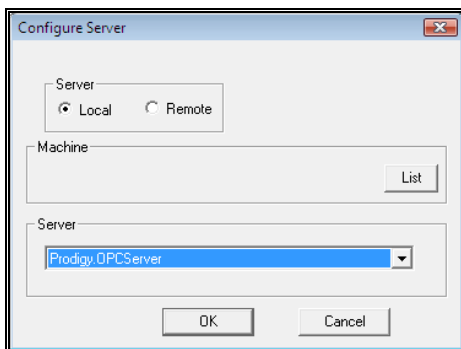
This is a fully functional OPC Client that will connect to any OPC Server to read and write OPC Items. It allows the user to select Servers, create groups and add items to these groups. The Client displays information about groups and items and updates as values change. Configurations of Servers, Groups and Items may be saved for loading it at a later date but no values are saved.

You can load in an existing configuration by using the Open icon or menu item.

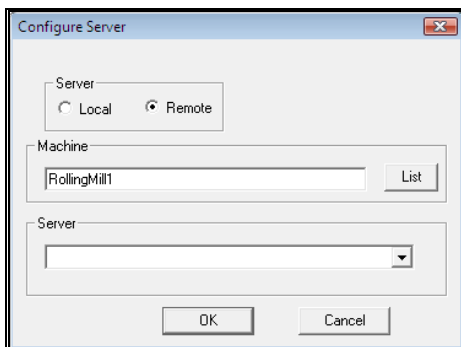
To begin without an existing configuration, click on the Connect to Server icon.



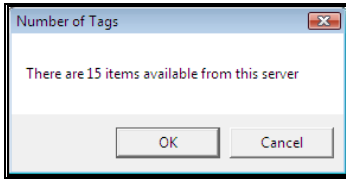
Select **Local** or **Remote** as appropriate.



If you select **Remote** you will have to enter the name of the remote machine that you want to scan for servers.



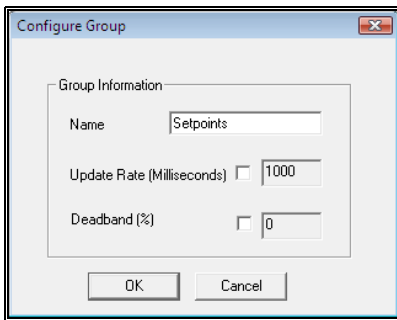
Click the **List** button. This will list all of the servers currently registered on the specified machine into the **Server** dropdown list. Select the server you require. You will be told how many OPC Items are available from this OPC server.



Next an OPC group must be added by clicking the **Add a Group** icon.



This option is only available when the server is the highlighted text on the left-hand side. You will be presented with the following dialog.



- Enter a name for the group.
- Enter an **Update Rate**.
 - If you want to modify the rate from the default check the box and enter the desired rate.
 - All signals in the same group will have the same update rate.
 - When a server is connected to, it will check whether the requested update is achievable. If it is not it will respond with the best rate that it can achieve and a warning message will be placed in **Spy**.
- Enter a **Deadband**.

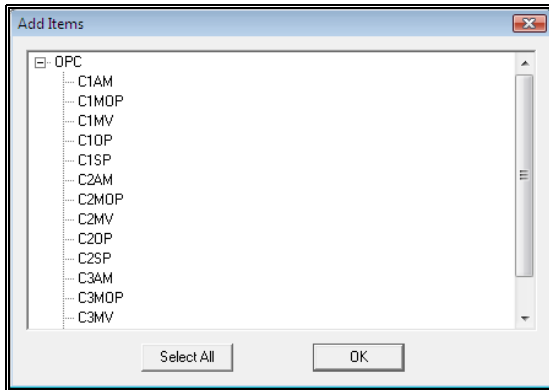
The deadband specifies how much the signal must change by before the server notifies the client that it has changed. At the default value of 0 all changes will be notified.

 - If you want to modify the deadband from the default check the box and enter the desired deadband.
 - All signals in the same group will have the same deadband.

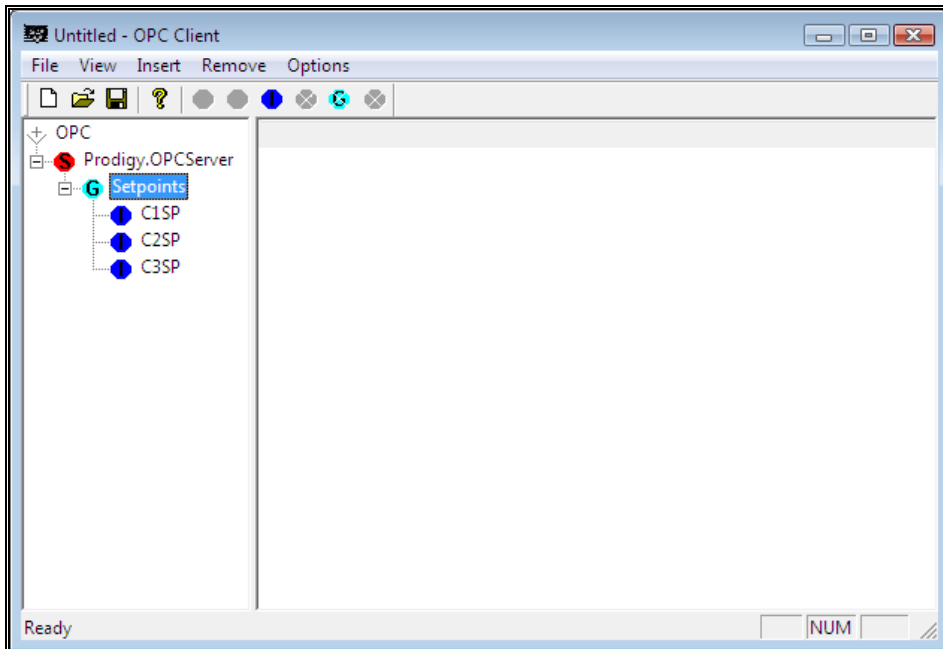
Once a group has been added you can begin adding items to the group by clicking the Insert Item icon.



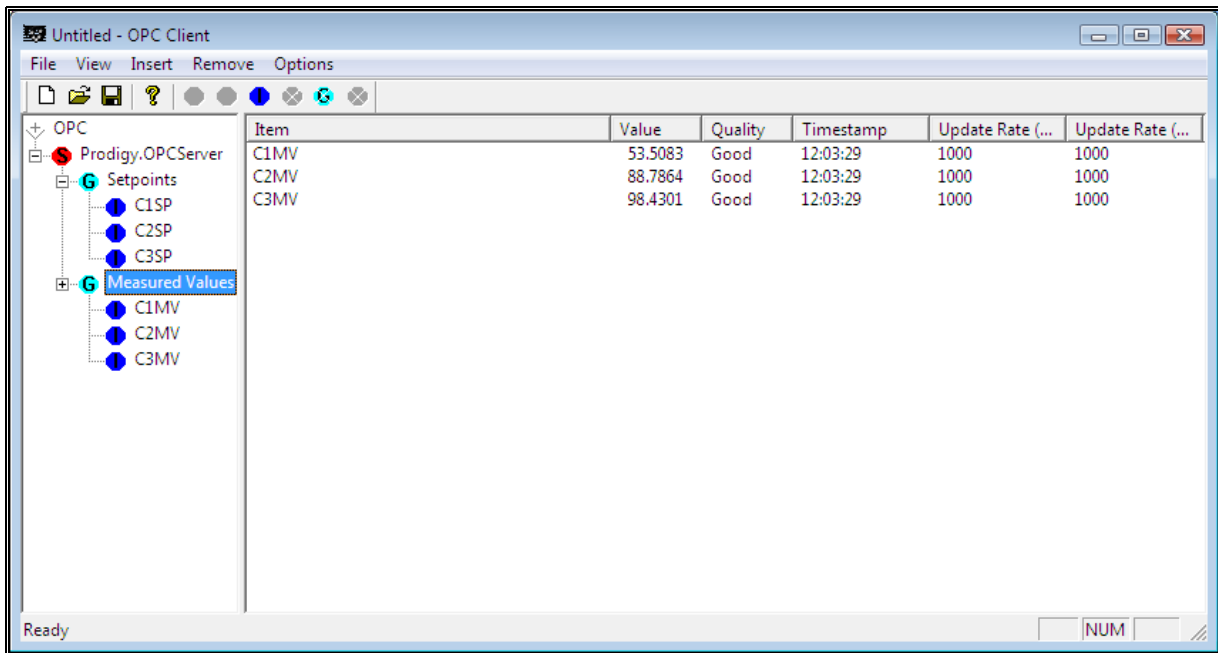
This option is only available when a group is the highlighted text on the left-hand side. The following dialog is displayed which lists all of the items available from the specified OPC server.



Double click on an item to add it to the current group. Alternatively click **Select All** to add all items to the group. In our example the group is called *Setpoints* and only the setpoint signals (ending with **SP**) will be added (see below).

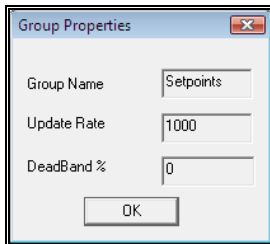


When you click on the group names on the left-hand side all of the items within this group are displayed on the right hand side, along with some diagnostic information

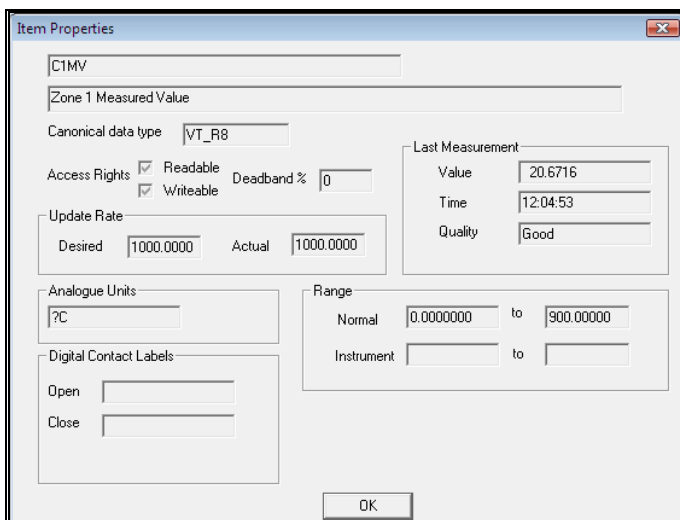


The display will update when the values change if there are a small number of items. If there are many items displayed the display will update every few seconds.

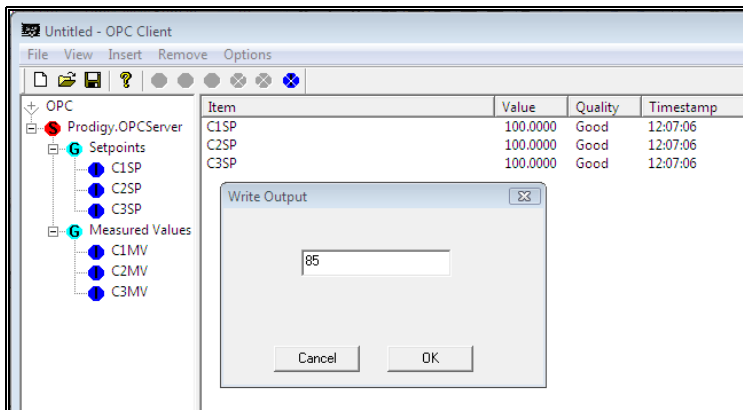
If you click on a group in the left-hand pane and then right-click it, you will see the properties for that group.



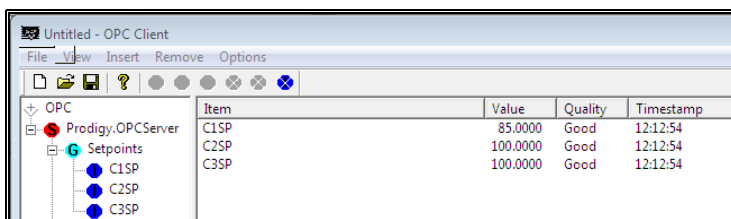
Similarly, clicking and then right-clicking an item in the left-hand pane will display the properties for that item. A minimum number of properties will always exist, such as Item name, Access Rights and canonical data type (this is the type the item is stored as, for example VT_R4 is a 4 byte real, and VT_I2 is a two byte integer). Other properties may exist depending on the implementation of the specified server.



Item values that are defined by the server to be writeable may have values sent to them from the client. Do this by double-clicking an item in the left-hand pane to display the following dialog:



Enter a value and click OK to send it to the server.



Servers, Groups and Items can be deleted by clicking the **Delete Server**, **Delete Group** and **Delete Item** icons (below).



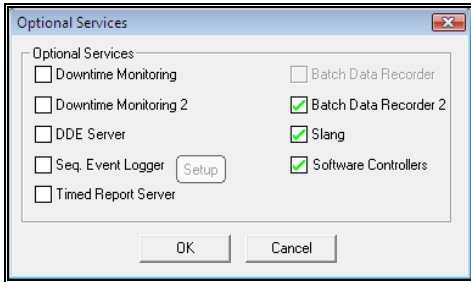
Configurations may be saved and loaded into the client at a later date using the usual **Save**, **Save As** and **Open** facilities common to most Windows programs.

DDE

DDE (Dynamic Data Exchange) allows you to send "live" values from Prodigy to other applications such as Excel.

Prodigy DDE Server

To enable the DDE server in Prodigy check the DDE box in the Prodigy Control Panel Optional Service.



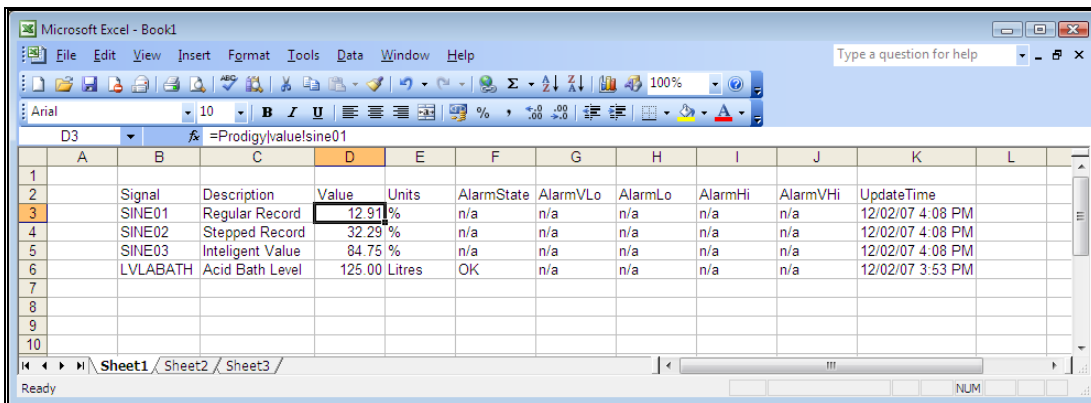
The software you use to get the DDE values from Prodigy (e.g. Excel™) will require you to enter information on the **application**, **topic** and **item**. these are:-

Application - prodigy

Topic - Description, Units, AlarmState, AlarmVLo, AlarmLo, AlarmHi, AlarmVHi, UpdateTime, Value

Item - any existing signal name (e.g. MYSIGNAL)

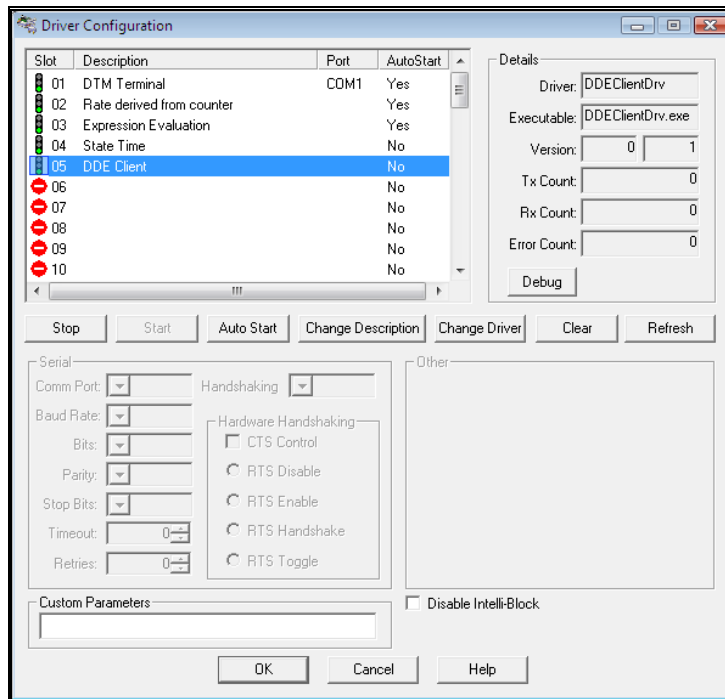
e.g. to display the value of signal SINE01 in Excel, enter "=prodigy|value!sine01" as the formula of a cell, as shown below:



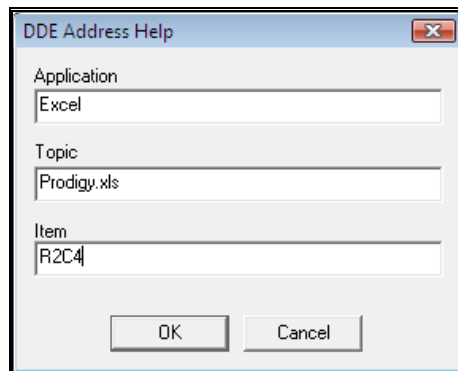
Prodigy DDE Client

This allows Prodigy to act as a client to a DDE server.

- Add the DDE Client driver into a driver slot using the Driver Wizard.



- Configure your signal in the Signal Database Configuration screen to use the DDE Client driver, and in the address configuration insert the Application, Topic and Item names. e.g.
 - Application **Excel**
 - Topic **prodigy.xls**
 - Item **R2C4** would connect the signal to the cell at row 2, column 4 of the Excel spreadsheet "prodigy.xls".



What is a Controller

Three Term or PID (Proportional-Integral-Derivative) controllers allow a process to accurately achieve and maintain a required setpoint by adjusting the output level. Many different manufacturers provide such devices in the form of hardware modules and in addition, Prodigy provides the facility within a software module.

How PID Control Works

When there is a difference between the measured value and the setpoint the controller must adjust the output power to compensate for this change. In determining the output power required three factors are taken into account, and these are the three terms of the PID controller. The actions are described by way of example below.

To ease the explanation we will assume that the required setpoint is higher than the current measured value. If it were lower the PID controller would work in much the same way but with some of the actions reversed. This will be explained a little better at the foot of the example. The example assumes that the controller is operating with the proportional term being proportional on error (default setting); this is the mode that the auto-tune facility can calculate terms for.

- **Proportional Term.**

The proportional term contributes an amount to the output power proportional to the error between the setpoint and measured value. The response of the proportional term can be adjusted by changing the value of the proportional gain. A high proportional gain will result in a large response to a small error. For example an error of 1°C with proportional gain of 1 will result in the proportional term contributing 1% to the power, where as if the proportional gain is 10 the controller will contribute 10% to the output power.

- **Integral Term.**

The integral term takes into consideration the error over time. As indicated by its name, the term integrates the error with respect to time giving a running total of the error that is added to the output power. With each consecutive iteration of the controller algorithm an additional amount of power is added to the integral term and hence to the output power. This additional power will eliminate any steady state error that would be present in just a proportional controller and will provide all the power output that the system needs once it reaches a steady state.

The error that is added to the integral term is multiplied by the integral gain. The equation for the amount that is added to the integral term is:

$$IG * (\text{Error} + \text{Last Error}) / 2 * dT$$

Suppose that:

- The difference between the times of two successive readings (dT) is 1 second.
- The Error is 5° and the Last Error is 15°.
- The Integral Gain (IG) is 0.1.

The equation is $0.1 * (5 + 15) / 2 * 1 = 1\%$, so the integral term will provide an additional 1% to the output power. If the same is true for each iteration of the controller the output power will increase by 1% per second until the total output power reaches 100% or a steady state is found. If the integral gain is double to 0.2, 2% would be added to the power per second and if it were halved 0.5% would be added per second. Tailoring this integral gain can affect the speed at which setpoint is achieved (power is more rapidly or less rapidly applied) and can also affect the potential overshoot (a large integral term takes longer to unwind).

If the integral gain is too large oscillation will occur about the setpoint as the integral term will wind up and down very quickly which would result in control similar to on/off control about the setpoint.

- **Derivative Term.**

The derivative term takes into consideration how quickly the error is changing. The magnitude of the derivative gain will determine when the derivative term begins to counteract the proportional and integral terms to pull power off before the setpoint is reached, in an attempt to avoid a large overshoot. If the measured value is changing quite slowly derivative action will be minimal, but as the rate increases, so derivative action increases to effectively reduce the output power and slow the rate of change of the measured value.

In theory the derivative term takes into consideration the rate of change of the error and adds this to the controller output power. It needs to be noted that at the time the setpoint changes the rate of change of error can be extremely high and this can cause a “kick” in the output power of the controller, which is generally undesirable. It can be proven that the derivative of the error is the same as the negative derivative of the measured value except when the setpoint changes.

As $e(t) = SP - MV$ it can be shown that:

$$\frac{de(t)}{dt} = \frac{d(SP - MV)}{dt} = \frac{d(SP)}{dt} - \frac{d(MV)}{dt}$$

As the setpoint (SP) can be considered to be constant whilst the controller is trying to achieve a setpoint and the derivative of a constant is 0 it can be shown that:

$$\frac{de(t)}{dt} = -\frac{d(MV)}{dt} \quad \text{as} \quad \frac{d(SP)}{dt} = 0$$

This means that rather than adding the rate of change of the error for the derivative term to the output power, the rate of change of the measured value is subtracted from the proportional term and the integral term. Doing this avoids spikes in the output power of the controller.

In a best-case scenario, the Derivative will allow the temperature to rise rapidly until it approaches setpoint. At this point the derivative kicks in and starts to pull the output power back before any overshoot occurs.

The derivative gain is a number that is used to factor the derivative term and is related to the rate of change of the slope over a period of time. For example if the derivative gain was 10 and the rate of change of a measured value was 0.1° per second then a value of 1% would be subtracted from the output power.

The derivative term also helps reduced disturbances that can occur in a system. For example say the door on an oven was opened causing the temperature to drop suddenly; the derivative term will very quickly add power into the system to compensate for this drop in temperature and help reduce the amount by which it would drop if there was only the proportional term and the integral term there to combat the disturbance.

Therefore the total output power of the controller is equal to:

Proportional Term + Integral Term – Derivative Term

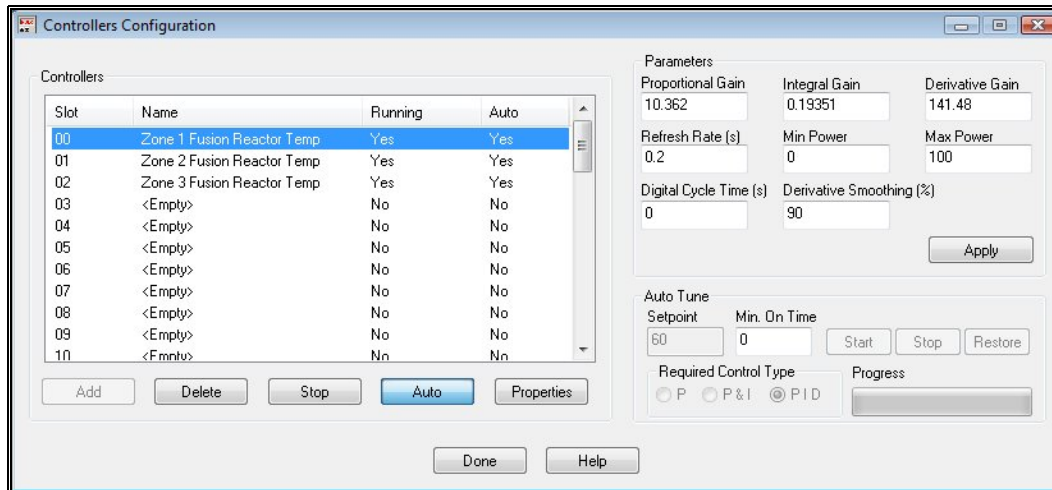
The ideal operation of a PID controller where the setpoint is increased by a substantial amount is as follows:

- There is a large error and the proportional term increases the power to 100% instantly causing the measured value to increase towards the setpoint at the maximum rate possible.
- As the measured value approaches setpoint the contribution of the proportional term will decrease. At this point the integral term will begin to increase its contribution and keep the power level sufficiently high to approach setpoint.
- As the measured value increases the derivative term increases, as the measured value gets near to the setpoint the proportional term will eventually be smaller than the derivative value. At this point the power will be reduced by the controller.
- The thermal inertia in the system should continue to cause the temperature to increase until it reaches setpoint. When this occurs the integral term should have a value that is high enough to apply the right amount of power to hold the measured value at the required setpoint.

Configuring the Controllers

Prodigy controller configuration works in a similar way to driver configuration. Controllers are allocated to *slots* and there are currently 40 slots available.

Adding, Removing, Starting and Stopping



Each slot shows the name of the controller that is allocated to that slot or <Empty> if there is no controller. It also shows whether or not the controller is currently running and whether it is set to auto-run.

- To stop a controller that is currently running, click on the **Slot** in the list and then press the **Stop** button. The button will change to show **Start** and the text under the **Running** column will change to show **No**.
- To start a controller that is not running, follow the same procedure, but the button will read **Start** instead of **Stop**.
- To set a controller to auto-run, (that is, when the services start up, that controller will be automatically started) select the required slot and press the **Auto** button. The button will stay indented and the text under auto will read **Yes**.
- To stop a controller from being auto-run, follow the same procedure, but the button will be indented to begin with and will not be indented after clicking on it.
- To add a new controller select an empty slot and press the **Add** button. This will create an entry in the slot with a default name. Fill in the relevant parameters for the controller (by clicking the **Properties** button) before **starting** it.
- To delete a controller, select the required slot and press the **Delete** button. The controller will be stopped if it is running and its configuration will be deleted.

The fields on the main screen show the current state of some of the key parameters that can be set for a controller. The values can only be changed in two ways; first by pressing the **Properties** button then changing the appropriate values and clicking **OK** or secondly by entering a new value in the parameters section and then pressing the **Apply** button.

The group of objects in the Auto Tune segment can be used to obtain values for the P, I and D terms; performing an auto tune is described later in this section of the manual.

Setting the Parameters

To change the parameters of a controller, select the required controller from the list and press the **Properties** button. The following dialog will be displayed.

The screenshot shows a 'Parameters' dialog box with the following fields and options:

- Title:** Zone 1 Fusion Reactor Temp
- Control:**
 - Measured Value: C1MV (Select...)
 - Setpoint: C1SP (Select...)
 - Refresh Rate (Secs): 0.2
- P.I.D. Terms:**
 - Proportional Gain: 1 (Select...)
 - Integral Gain: 0.8 (Select...)
 - Derivative Gain: 1 (Select...)
 - Derivative Smoothing (%): 0
 - Direct acting (More power cools/slows):
 - Proportional On Setpoint:
- Digital Control:**
 - Viewed O.P.: (Select...)
 - Cycle Time (Secs): 0
- Power:**
 - Signals: Auto/Manual
 - Output power: C1OP (Select...)
 - Man. O.P.: C1MOP (Select...)
 - Output Power Range:
 - From Signal:
 - Min Power: 0
 - Max Power: 100
 - Bumpless Transfer:
 - Manual to Auto:
 - Auto to Manual:

Buttons: OK, Cancel, Help

The various groups and fields are as follows:

- **Title.**
This is a name for the controller and is used merely as an aid to identification.
- **Control.**
 - **Measured Value.**
Press the **Select** button to enter the signal that is being used for the measured value of the process being controlled. This should be an analogue signal.
 - **Setpoint.**
Press the **Select** button to enter the signal that is being used for the setpoint of the process being controlled. This should be an analogue signal.
 - **Refresh Rate.**
Enter a value that is the time in seconds between successive calculations of the output power. In other words, the controller will recalculate the output power every **n** seconds, where **n** is the number you enter here. This number should be in the range 0.1 to 300 seconds – pick a sensible value for the process being controlled.
- **Digital Control.**
Sometimes the device being controlled cannot respond to an analogue value but can only be **on** or **off**. In order for our controller to achieve say 40% output power it must turn the device **on** for 40% of the time and **off** for 60% of the time. The fields in the **Digital Control** group only become available if a digital signal is selected for the **Output power** in the **Power** group. The two fields in the **Digital Control** group are
 - **Viewed O.P.**
The viewed output power is the output power that the controller is trying to achieve. As the signal for output power is digital and is therefore only ever 1 or 0, it would not be possible to show (on a Display for example) the actual output power of the controller. The Viewed O.P. signal (which must be an analogue) can be used to display this if required.
 - **Cycle Time.**
As mentioned earlier, in order to achieve 40% power the controller would turn the device on for 40% of the time and off for 60% of the time. The time period in question is the Cycle Time. If a

cycle time of 10 seconds is given, the device would be on for 4 seconds and off for 6.

Great care must be taken when selecting a cycle time. Although it is true that whatever value you choose would result in the same average power output, too large a number will result in wildly fluctuating readings. Consider the effect if our device was a heater and it was on for 40 minutes and then off for 60 as opposed to being on for 4 seconds and off for 6. It could get rather too hot during the 40 minutes it was on!

Conversely, too short a cycle time could result in a reduced lifetime for some components such as relays or switches. Choose the cycle time carefully to strike the correct balance between good control and unnecessary switching.

- **P.I.D. Terms.**

- **Proportional Gain.**

Press the **Select** button to enter the signal that is to be used to set the Proportional Gain for this controller. Alternatively, enter a numeric value into the field below this one.

The P.I.D terms can all be configured to come from signals, but it is most likely that fixed values will be used. The use of signals is best reserved for tuning the controller to find good values for P, I and D. The values thus obtained should then be entered as fixed values for P, I and D. Using signals will allow buttons on a Display to be configured to change the PID Gains rather than using the controllers window.

- **Proportional on Setpoint.**

When this button is checked the amount of output power contributed by the proportional term is based solely on the proportional gain and the value of the setpoint. This means that the proportional term is constant at a setpoint. If this button is not checked then the proportional term is calculated based on the error and the proportional gain; for most applications it is recommended that this option isn't checked and control is done based on the error.

- **Direct Acting.**

In a *direct acting controller* more output power causes the measured value to decrease (e.g. a refrigeration unit). Check this box if that is the action required, leave it unchecked if more output power causes a rise in measured value (e.g. a heater).

- **Integral Gain.**

Press the **Select** button to enter the signal that is to be used to set the Integral Term for this controller. Alternatively, enter a numeric value into the field below this one.

- **Derivative Gain.**

Press the **Select** button to enter the signal that is to be used to set the Derivative Term for this controller. Alternatively, enter a numeric value into the field below this one.

- **Derivative Smoothing.**

Derivative smoothing is available to improve the effectiveness of the derivative term. For example, suppose a temperature is increasing by 1° every 10 seconds and the resolution of the device reading the temperature is only 1° (so temperatures appear as integer values). In any 10 second period the temperature will appear to be constant for most of it before suddenly, and seemingly instantaneously, jumping up by 1° (e.g. 15° for 9.99 seconds before jumping to 16° in the following .01 seconds). This gives a derivative slope of 0 for the 9.99 seconds and a very steep slope for the remaining fraction, resulting in very uneven derivative action. Smoothing overcomes this.

The value entered in the smoothing field should be between 0 and 99.9. If 90 was entered then 90% of the previous derivative value would be held on the derivative term and 10% of the new derivative value added on. This would slow the response of the derivative term slightly but would mean that the effect of the derivative term would be seen over several refresh periods rather than just an instantaneous blip to the power.

- **Power.**

- **Auto/Manual.**

Select a digital signal that determines whether or not the controller is in automatic or manual mode. When the controller is in **automatic** mode it calculates the required output power based on the methods described previously. When it is in **manual** mode it does no calculations but

simply uses the value of the **Manual O.P.** signal as the value of the output power. If **Digital Control** is being used the output power will be converted to the correct digital state based on the **cycle time** even when the controller is in manual mode.

A value of 1 in the Auto/Manual signal sets the controller into automatic mode and 0 sets it to manual.

- **Output Power.**

Select the signal that is to be used for the output power as calculated by the controller. This signal can be either analogue or digital and if it is digital the fields in the **Digital Control** group will become available.

- **Man O.P.**

Select the signal that is to be used for the manual output power. This signal must be analogue. When the controller is set into manual mode (via the **Auto/Manual** signal) the value of this signal is copied to the output power signal (taking into account any **Digital Control** required).

- **Output Power Range.**

The range of the output power can be taken either from the range of the signal that you select for output power (this range is defined in **Database Configuration**) or it can be entered manually here. To use the range from the signal definition, check the box labelled **From Signal**. The signal range will be displayed in the **Min Power** and **Max Power** boxes, which will become read-only.

Although the output power may often be 0 to 100%, it need not always be so. It may be that a cooling fan can have its actual speed of rotation set, say in the range 0 to 4000rpm. The Power range is taken into account when performing an auto tune so changing this range will require the controller to be re-tuned.

- **Bumpless Transfer.**

- **Manual to Auto.**

When this button is checked bumpless transfer is used when switching from manual operation to automatic operation. A Controller Output Power Bias equal to the last manual output power is added to the controllers output power to ensure that the transition to Auto mode is as smooth as possible. If this button is not checked then when the controller is put into automatic mode the output power will drop to zero and then change to hold at the required setpoint.

If the system is in a steady state in manual operation and is switched to automatic operation the controller will react and change the output power if the controller setpoint isn't equal to the steady state measured value.

- **Auto to Manual.**

When this button is checked bumpless transfer is used when switching from automatic operation to manual operation. The manual output power signal is set to be the same value that the controller was outputting when in automatic mode so the output power stays constant. If this button is not checked then when the controller is put into manual mode the output power will be set to value held in the manual output power signal.

Applying or Cancelling

When you press the **OK** button all of the data entered onto this screen is passed directly to the controller and has immediate effect. In addition, it is saved to a configuration file to be read in next time the controller driver is started. If you press the **Cancel** button, the changes are discarded.

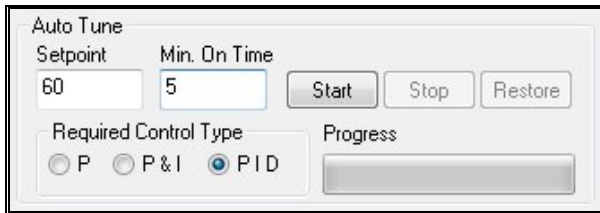
Auto Tuning a Controller

Prodigy controller configuration comes with an Auto Tune facility. The auto tune algorithm will provide P, I and D terms that will give reasonable control of a system. If the values that are obtained from an auto tune give acceptable control then they can be used as is, but if the control is not within the required limits the P, I and D terms can be manually adjusted in an attempt to improve performance.

In order to get the best results from auto tuning the starting conditions and the setpoint should reflect as accurately as possible the conditions under which control is normally performed. If it is usual to control an oven that starts from ambient and goes to 220°C then it is better to auto tune with a starting temperature of ambient rather than, for example, 120°C. This can make repetitive tuning attempts time-consuming as the oven has to cool back to ambient before running another auto tune, but the end results will be better.

Auto tuning takes the form of a number of phases where power is applied to the system until it reaches setpoint and then removed until it falls back below the setpoint, at which point power is reapplied and the cycle repeats. As such it can be quite time consuming in systems where setpoint takes a long time to achieve. The algorithm is described in more detail further on in this section.

The Auto Tune Parameters

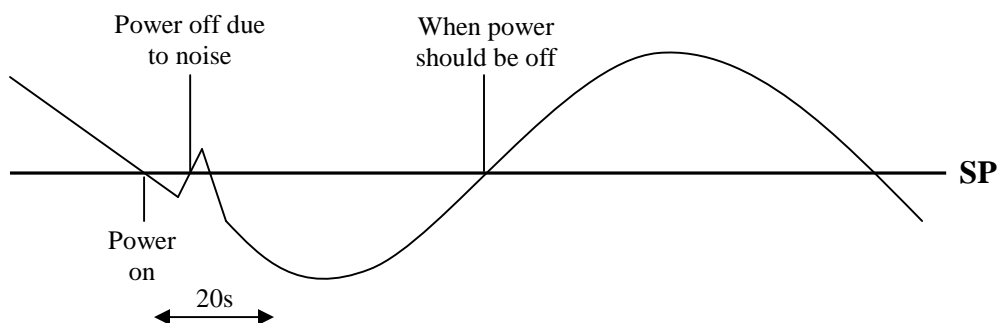


The various objects are as follows:

- Setpoint.**
 This is the setpoint about which an auto tune is performed, for the auto tune to give the best possible control when the system is running the setpoint that is picked should be around a very commonly used setpoint. For example if an oven runs at 200°C most of the time and it's output is only varied between 170°C and 230°C then the setpoint should be 200°C when the auto tune is performed.

- Min On Time.**
 If there is noise on the measured value signal that may cause it to flicker above and below the setpoint quickly this will disrupt the auto tune algorithm and provide incorrect P, I and D terms. To avoid this the Min On Time field can be set to a value in seconds that the auto tune algorithm should keep the power on or off for.

If there is a lot of latency in a system, for example in an oven the heating may be applied but it may take 30seconds for the heating to have any affect, during which time the overall trend of the measured value may be a drop in temperature. As the measured value drops below the setpoint the heating will be turned on; noise may cause it to rise above the setpoint which will cause the heating to be turned off. This will count as two phases in the auto tune when really it should only be one; to avoid this situation a **Min On Time** of 20 seconds could be applied.

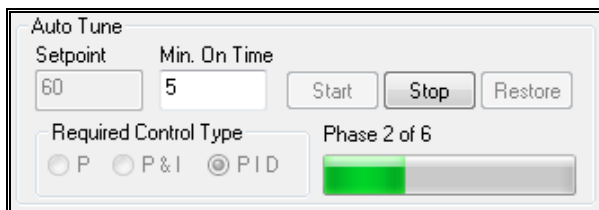


As can be seen in the diagram above the 20s delay would result in the noise spike (exaggerated) being ignored by the auto tune algorithm and the correct P, I and D terms would be calculated.

- Start.**
 Pressing this button will start an auto tune about the currently entered setpoint.
- Stop.**
 Pressing this button will stop an auto tune that is running; the output power will be set to 0 if the stop button is pressed.

- **Restore.**
If an auto tune has been performed on a system in the past pressing this buttons will restore the last values that were obtained from that auto tune for the P, I and D terms. This is a useful function if the operator has modified the controller terms and cannot remember what the auto tune ones were, as it saves time as another auto tune does not need to be performed (an auto tune could take a matter of seconds, but more probable is it would take minutes or even hours).
- **Required Control Type.**
These are the three types of auto tune that are available in the Prodigy software controller.
 - **P.**
This is a Proportional only controller; this type of control will probably end up with a steady state error.
 - **P & I.**
This is a Proportional and Integral controller, this type of control will have no steady state error but will probably overshoot and have a long settling time.
 - **PID.**
This is a three term controller, in most situations this will provide the best control with no steady state error, a reduced overshoot and reduced settling time. However if there is lots of noise on a signal the derivative term could cause the system to be unstable and the output power to vary rapidly.

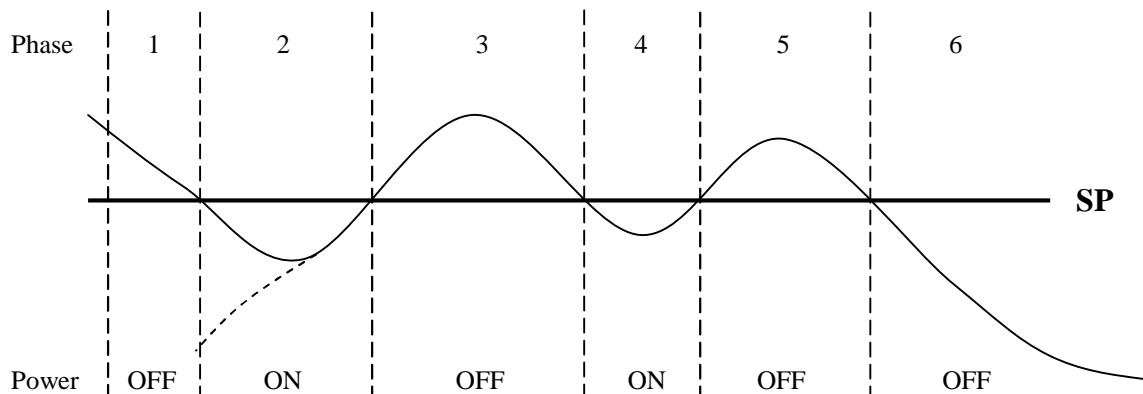
Performing an Auto Tune



In order to perform an auto tune the controller must be stopped, if it is running the fields and buttons in the above section will be read only (with the exception of Restore).

To begin select an appropriate Setpoint, the Required Control type and a Min On Time (if required). Now press the start button, the progress bar will display what phase the auto tune is in, if the Measured Value is plotted on a trend it should appear as below in some distinct phases and the power will be On and Off in each phase as can be seen (on represents 100% power and off 0% power).

This example assumes that the system been auto tuned is a heater that is Reverse Acting (PowerOn = Heating) and that the system is proportional on error (not setpoint).



- **Phase 1.**
Phase 1 is the preliminary cooling cycle. This will only occur if the measured value is greater than the setpoint when Start is pressed. If this is the case the auto tune algorithm will wait until the

temperature drops below the setpoint, at which point the power is turned ON and the algorithm moves into Phase 2. If the measured value is already below setpoint then Phase 2 begins immediately.

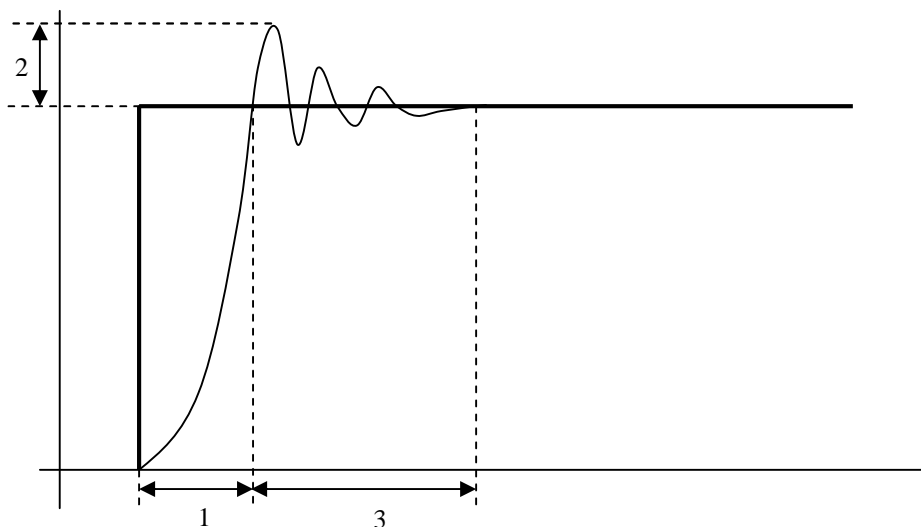
- **Phase 2.**
Phase 2 is the primary heating cycle; the power is ON until the measured value increases to a value greater than the setpoint. When this occurs the power is turned OFF and the algorithm moves into phase 3.
- **Phase 3.**
Phase 3 is the primary cooling cycle; the power is OFF until the measured value drops below the setpoint. When this occurs the power is turned ON and the algorithm moves into phase 4.
- **Phase 4.**
Phase 4 is the measurement heating cycle; measurements of time and the minimum measured value are taken whilst the power is left ON. When the measured value is greater than the setpoint the power is turned OFF and the algorithm moves into phase 5.
- **Phase 5.**
Phase 5 is the measurement cooling cycle; measurements of time and the maximum measured value are taken whilst the power is left OFF. When the measured value is less than the setpoint the power is left OFF, measurements are stopped and the algorithm moves into phase 6.
- **Phase 6.**
Auto tuning is complete. The system is left to cool to ambient temperature and calculations are performed on the measurements taken during phase 4 and 5. From these calculations P, I and D Gains are calculated for use in the controller.

The auto tune algorithm uses the latency in systems to allow it to operate and calculate the P, I and D gains. So for example if an oven system applies power and takes 1 minute to react, but the auto tune only takes 3 minutes to complete, then the auto tune probably hasn't been effective (probably due to noise on the measured value). The operator should look into increasing the Min On Time or trying to reduce noise on the measured value signal.

Manually Modifying the P, I and D Gains.

The auto tune algorithm is designed to give a P, I and D gains that give reasonable control to a system. After the auto tune values have been obtained they can be used as a basic guide as to the figures the controller requires and further tuned manually to improve the performance of the controller.

Below is a diagram of a trace, the thick line is the setpoint; it has undergone a step change, the thin line is the measured value and how it reacts to the control applied to the output power.



The three numbers on the above diagram relate to characteristics of a measured value under control these are the Rise Time (1), the Overshoot (2) and the Settling Time (3). The table below shows the effects on the three characteristics plus the Steady State Error (S.S. Error) of increasing the parameters.

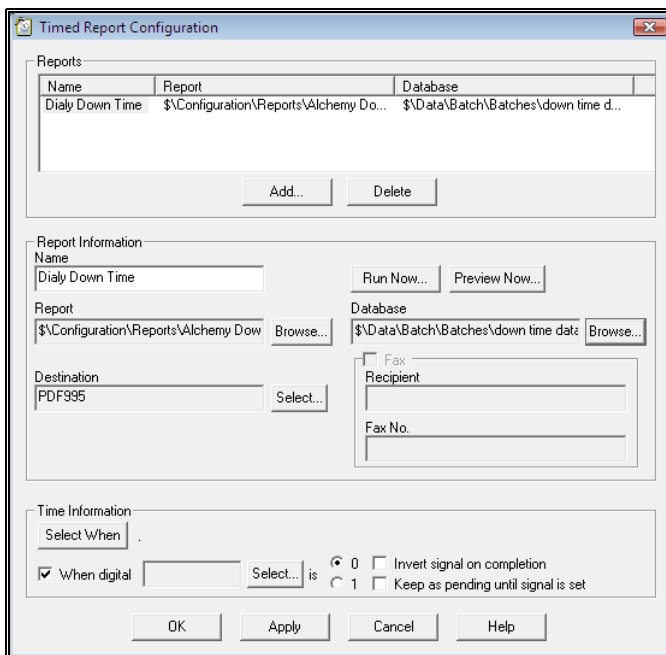
Parameters	Rise Time (1)	Overshoot (3)	Settling Time (3)	S. S. Error
P	Decrease	Increase	Small Change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Slight Increase	Decrease	Decrease	None

Timed Reports

What Are Timed Reports

Prodigy provides a facility whereby reports can be run automatically at specified time intervals. This facility could be used to print reports at the start or end of each day, each shift or every hour, for instance. The timed report facility consists of two complementary parts, Timed Report Configuration and the Timed Report Server. The Timed Report Server is always running as part of the standard Prodigy Services; whether or not it does any work depends on whether any timed reports have been configured.

Timed Report Configuration



At the top of the dialog is a list of all of the reports that have been configured. When an item is selected from this list the fields below become active and show the relevant information for the selected report.

The fields and buttons on the dialog are as follows

- **Reports.**
 - **Add.**

Adds a new entry into the list. You must subsequently enter information into the fields below to properly name and configure this report.
 - **Delete.**

Deletes the currently selected report.
- **Report Information.**
 - **Name.**

This is the name you wish to give to this report. It has no other purpose than being a label for this entry in the Timed Report list. You may use the same actual **Report** (see below) several times with different names, e.g. a report called Shift Totals may be used 3 times as **Hourly Shift Totals**, **Daily Shift Totals** and **Weekly Shift Totals**, the only difference being the information saying when the report is to run. The name may be used when reporting errors, e.g. “Weekly Shift Totals failed to run”.
 - **Report.**

This is the actual report that is to be run. Use the **Browse...** button to select the required report.

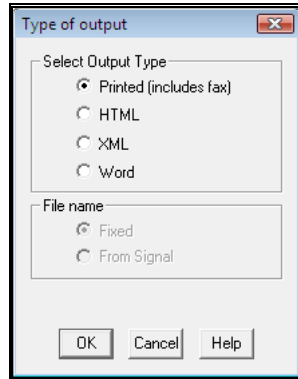
- **Database.**

This is the database that is to be reported on. Use the **Browse...** button to select the required report.

- **Destination.**

This selects the format and destination of the output from the report. The report can be sent straight to a printer or a fax modem, or it can be rendered into HTML format for displaying in a web browser.

Click the **Select...** button to display the following dialog.



- If you choose **Printed** you will then be presented with the standard printer selection dialog. From within the printer selection dialog, choose the **Properties** option to set the required print options, e.g. paper orientation, print quality, etc. If using **Faxing** (see below) select the appropriate fax renderer.
 - If you choose **HTML** you will also need to choose whether the report will be placed into a fixed file name and location, or whether this will be taken from a signal.
If you choose **Fixed** you will be asked to provide a full path name for the file.
If you choose **From Signal** you will have to choose a directory into which the file will be placed, and a signal name that will provide the name of the html file. The signal should be a derived digital signal.
See HTML Reports and Web Upload for more details.
 - If you choose XML it works exactly as for **HTML** and **Word** but produces output in XML (Extensible Mark-up Language) format.
 - If you choose Word it works exactly as for **HTML** and **XML** but produces output in Microsoft Word™ format.
 - **Run Now.**
This button causes the report to be run immediately, regardless of any time or signal constraints. Output goes directly to the destination selected in the **Destination** field.
 - **Preview Now.**
This runs the report immediately and displays the output in a preview window on the screen.
 - **Faxing (Optional Facility)**
If a faxing facility has been provided with Prodigy then the **Fax** check box and its associated options will become available. These fields are as follows.
 - **Recipient.**
This is the name of the recipient of the fax and is used for information purposes only.
 - **Fax No.**
This is the full telephone number of the recipient of the fax and should include any code necessary to get an external line (usually 9).
- In order for the report to be converted into a format suitable for faxing it must be **rendered** and this is done by selecting a fax renderer as the **Destination**. Prodigy currently uses **32BitFax** as its faxing software and this comes with a renderer called **32BF NT Convert to Fax**. This must be selected from the dialog displayed when the **Select...** button is pressed (see above).
- When a timed report is run and **Fax** is its destination the following actions occur.

- The report is run in the background and output directed to a printer, as for non-fax reports. In the case of a fax though, the printer is actually a fax renderer that produces a bitmap file that can be transmitted to a receiving fax machine.
- After the bitmap file is produced Timed Report Server passes the recipient information (name and telephone number) to the faxing software.
- The faxing software uses this information to connect (via a suitable modem) to the receiving fax machine.
- The bitmap file is transmitted.
- All relevant logging of information takes place and can be viewed from within 32BitFax.

*Prodigy faxing is an optional facility and is written to use **32BitFax** at present and will not work with other faxing software that you may have. If you require faxing from within Prodigy then please contact your supplier who will be able to provide and configure 32BitFax.*

- **Time Information.**

- **Select When.**

This allows you to specify the time at which the report is to be run. Clicking the button displays a dialog that allows you to specify the time at which the report is to be run.

- **When Digital.**

This allows the running of the report to be not only dependent upon the time, but also upon the condition of a digital. Use the **Select...** button to choose the required signal and then select either the **0** or **1**. When the correct time comes around the Timed Report Server will check if the value of the signal is equal to the value specified here. If it is the report will be run; if it is not, one of two things will happen.

- If **Keep as pending until signal is set** has been checked, the report will be held waiting to run until the signal is set to the required value. As soon as this happens, regardless of the time, the report will run. If sufficient time elapses that the reports time condition is satisfied again (from the example dialog above, if an hour elapses the report will be due to be run again) the report simply stays in the queue - Timed Report Server will not run the report twice.

*An example of when to use the **When digital** and **Keep as pending until signal is set** options. Say you are reporting hourly on the level of fluid in a tank that may be agitated. It may be inappropriate to report on the level while the agitator is running, so a digital is used to indicate when agitation has been stopped for sufficiently long for the level reading to be accurate.*

- If **Keep as pending until signal is set** option has not been checked the report will simply be discarded for this instance.
 - **Invert signal on completion.**
If this box is checked the signal that determines whether or not the report can run will have its value inverted after the report has been run.

Calibration

Calibration

Input and Output Signals may be calibrated within **Signal Database Configuration**. However this can be tedious and slow if many different signals are involved. An alternative method is to use **Prodigy Calibration**.

The screenshot shows the 'Calibration' window in Prodigy software. The window title is 'Calibration' and it has a menu bar with 'Configuration File' and 'Calibration'. Below the menu bar is a toolbar with various icons. The main area displays a table titled 'Oven 13 Temperatures' with the following columns: Signal, Description, Low Range, High Range, Scaled, Units, Output, Raw, Low, Set, High, Set, Calibrate, Low Raw Value, High Raw Value, Check, Last Calibrated, and Notes. The table contains 14 rows of data for different temperature signals, including '13: Oven Front Temp', '13: Oven LHS Temp', '13: Oven Back Temp', '13: Oven RHS Temp', and several '13: Ply' temperatures, along with '13: Spare' signals. Each row has input fields for 'Raw', 'Low', 'Set', 'High', and 'Calibrate' values, and a 'Check' checkbox.

Signal	Description	Low Range	High Range	Scaled	Units	Output	Raw	Low	Set	High	Set	Calibrate	Low Raw Value	High Raw Value	Check	Last Calibrated	Notes
1TC01	13: Oven Front Temp	0	400	31.89	'F		77.85	32		400			83.25	17677.00	<input type="checkbox"/>	25/11/08	
1TC02	13: Oven LHS Temp	0	400	20.38	'F		79.56	23		54			80.58	92.58	<input type="checkbox"/>	25/11/08	
1TC03	13: Oven Back Temp	0	400	61.12	'F		81.01	45		60			77.73	80.79	<input type="checkbox"/>	25/11/08	
1TC04	13: Oven RHS Temp	0	400	40.54	'F		82.19	45		60			68.91	24.28	<input type="checkbox"/>	25/11/08	
1TC05	13: Ply 1A Temp	0	400	34.57	'F		83.07	34		400			55.58	17677.00	<input type="checkbox"/>	25/11/08	
1TC06	13: Ply 1B Temp	0	400	19.58	'F		83.67	15		20			52.25	86.54	<input type="checkbox"/>	25/11/08	
1TC07	13: Ply 2A Temp	0	400	32.73	'F		83.98	32		400			48.94	17677.00	<input type="checkbox"/>	25/11/08	
1TC08	13: Ply 2B Temp	0	400	33.04	'F		83.98	32		400			34.34	17677.00	<input type="checkbox"/>	25/11/08	
1TC09	13: Spare	0	400	33.09	'F		83.70	32		400			31.37	17677.00	<input type="checkbox"/>	25/11/08	
1TC10	13: Spare	0	400	33.35	'F		83.12	32		400			18.35	17677.00	<input type="checkbox"/>	25/11/08	
1TC11	13: Spare	0	400	33.29	'F		82.25	32		400			20.20	17677.00	<input type="checkbox"/>	25/11/08	
1TC12	13: Spare	0	400	33.31	'F		81.10	32		400			18.47	17677.00	<input type="checkbox"/>	25/11/08	
1TC13	13: Spare	0	400	33.31	'F		79.68	32		400			17.01	17677.00	<input type="checkbox"/>	25/11/08	
1TC14	13: Spare	0	400	33.25	'F		78.00	32		400			18.21	17677.00	<input type="checkbox"/>	25/11/08	

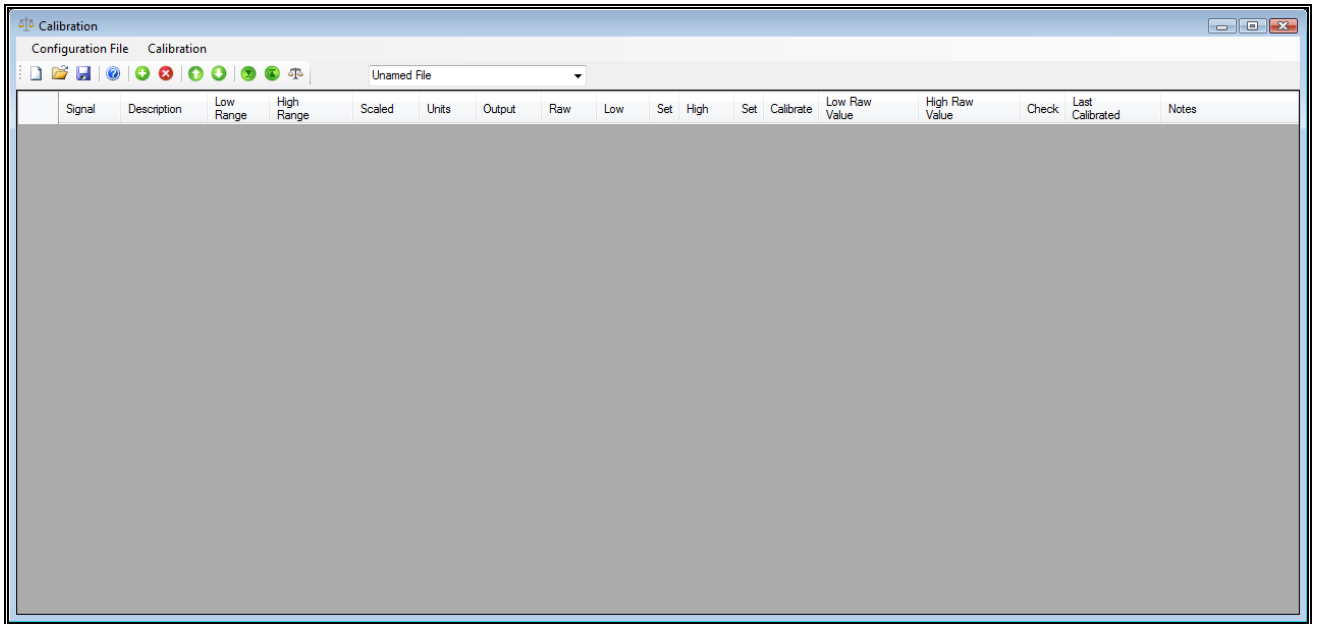
Prodigy Calibration allows a selection of signals to be calibrated using a named configuration and many configurations may be created to cater for different groups of signals. Each signal is calibrated by specifying two endpoint values in both External units (Raw Values) and Engineering Units (Scaled Values). All subsequent readings of this signal will be scaled using these calibration values. A user may check a calibration and attach notes to each signal if they wish.

Both the external and engineering values are stored in the Signal Database and also in an Access™ database, along with other calibration information, such as the Operator/User who performed the calibration, any special notes regarding a particular signal and calibration check values for the signals. Reports may be requested and displayed on the screen or printed showing either the calibrations on all signals between two dates or the calibration of one particular signal.


Creating a Configuration File

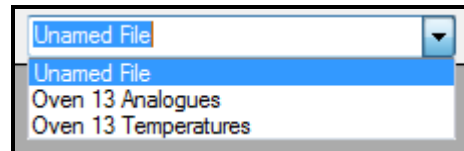
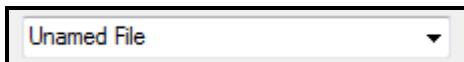
The calibration program may be invoked from a button on a Display or from the menu bar.

If the Prodigy Users facility is being used the user must have the *Run Calibration* privilege.



As a default Calibration loads in a blank configuration. Existing configurations can be selected through one of two methods.

- Pressing the open button will bring up a dialog that will show all current configuration files. 
- Using the drop down list on the right of the toolbar to select the appropriate configuration.



To add signals to a Calibration Configuration press the **Add Signals** button.



This will bring up the signal selection dialog with only signals that are suitable for calibration available. Selecting a signal will add it to the current configuration; multiple signals can be selected at once by holding down the shift key and clicking on the first and last signals that are to be added.

A signal can be removed from a configuration by selecting a signal and pressing the **Remove Signals** button.



The selected signal will be removed without a prompt so be sure you have the right signal selected when you press this button.

A signal's position in the calibration configuration can be altered by pressing the **Move Selected Signal Up** or **Move Selected Signal Down** button. These can be seen below.

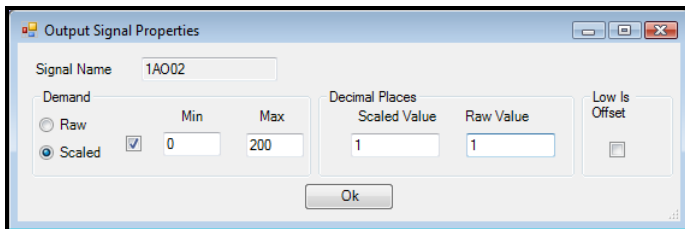
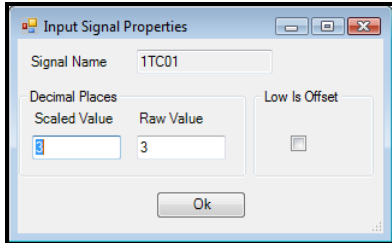


Move Selected Signal Up.



Move Selected Signal Down.

Once signals have been added to the calibration configuration options relating to each signal can be set by right clicking on a signal. One of two dialogs will be shown depending on whether the signal is an input or output signal. These two dialogs can be seen below.



The number of decimal places that the Configuration program will use when displaying the **scaled** and **raw** values can be specified on the dialog.

The **Low is Offset** button provides a custom offset single point calibration facility and is not normally used in the majority of situations.

Output signals can have a **Demand** specified; this is the output value that will be sent to a signal. The demand can be sent as either a **Raw** or **Scaled** value. If a scaled value is used then the previous or default **Scaled** and **Raw** values will be used to calculate and send a raw value to the output. To choose whether **Raw** or **Scaled** values are used select the appropriate check box.

Calibration can specify the minimum and maximum that the device can be set to by the **Calibration** software and is used to protect the external device from being overloaded during the calibration procedure. The **Min** and **Max** values can be set in either raw or scaled units. If you want to use limits, check the box, and enter the **Min** and **Max**, raw or scaled values.

Once the required selection of signals have been selected and their properties configured the calibration configuration can be saved. This can be done by pressing the **Save** button (below) or by selecting from the menu, **Configuration File** followed by **Save** or **Save As**.



To delete a Calibration Configuration select the configuration that is to be deleted. The configuration can then be deleted by selecting from the menu, **Configuration File** followed by **Delete**. A dialog will appear asking if the configuration is to be deleted and if yes is selected Calibration will return to a blank configuration.

Calibrating Signals

Once signals have been added into a calibration they can be calibrated using **Calibration**. To calibrate signals select the appropriate calibration configuration.

Signal	Description	Low Range	High Range	Scaled	Units	Output	Raw	Low	Set	High	Set	Calibrate	Low Raw Value	High Raw Value	Check	Last Calibrated	Notes
1TC01	13: Oven Front Temp	0	400	-136.395	*F		35.797	32		400			83.250	186.951		28/11/08	
1TC02	13: Oven LHS Temp	0	400	-85.225	*F		38.700	23		54			80.580	92.576		28/11/08	
1TC03	13: Oven Back Temp	0	400	-131.670	*F		41.748	45		60			77.730	80.785		28/11/08	
1TC04	13: Oven RHS Temp	0	400	53.065	*F		44.917	45		60			68.910	24.284		28/11/08	
1A001	13: Rig 1 Raise Pressure (SP)	0	100		psi	Set	0.0	0		20			0.0	20.0		27/11/08	
1A002	13: Rig 1 Lower Pressure (SP)	0	100		psi	Set	0.0	0		71			0.0	2500.0			
1A003	13: Rig 2 Raise Pressure (SP)	0	100	0.0	psi	Set		0		100			0.0	10.0			
1A004	13: Rig 2 Lower Pressure (SP)	0	100	0.0	psi	Set		0		72			0.0	2500.0		28/11/08	

Information shown by calibration is the signal name and description, the signal range (the range is used when displaying the signal on trends etc. but is not used as part of the calibration; it is here just as a convenience). The current scaled signal value is displayed, along with the units and the raw value. There is an output button that can be used to send a signal value to an output signal.

The columns labelled **Low** and **High** are where the engineering (or scaled) values should be entered. To enter values simply click on the column and type in the appropriate value. The values in the **Low** and **High** columns change colour to indicate their current state.

Low	Set	High	Set	Calibrate
45		60		
45	Set	60		Calibrate
23	Set	54	Set	
32		400		

If the values in the **Low** and **High** columns are grey then no modifications have been made to them, they are showing the previous scaled values. If a new Low or High value is entered in a column the text will turn blue and the **Set** button will become available (third Line). Pressing the **Set** button will turn the colour of the text red and remove the text from the **Set** button; the **Calibrate** button will become available. At the point when the **Set** button is pressed the current **Raw** value is recorded and linked with the Scaled value that was entered.

Pressing the **Calibrate** button will turn the text in the Low and High columns green, this will change the scaling factors within the calibration program but will not make changes to the values in the signal database. To save the new calibration values to the signal database the Save button on the toolbar must be pressed (note this saves any changes to the calibration configuration too) or by selecting from the menu, **Calibration** followed by **Save** or **Save and Exit**.

The remaining columns show the **Low** and **High Raw** values that are currently used to calculate the scaling factors for a signal. A **Check** button that can be used to check a signal once a calibration has been performed (described later). The **Last Calibrated** column that shows the date when a calibration was last performed and a **Notes** column that allows the operator performing a calibration to input any information he feels needs to be shown in relation to a performed calibration.

Low Raw Value	High Raw Value	Check	Last Calibrated	Notes
83.250	186.951		28/11/08	
80.580	92.576		26/11/08	
77.730	80.785			
68.910	24.284		28/11/08	

Calibrating Inputs

Set the input to some known low value, for instance for a temperature probe place it in iced water at 0 degrees. Wait until the current value reading settles and then enter 0 into **Low** and click **Set**.

Repeat the function for a known high value, e.g. temperature probe in boiling water and when the current value settles enter 100 into **High** and click **Set**.

Click the **Calibrate** button to make the required changes to the scaling factors used.

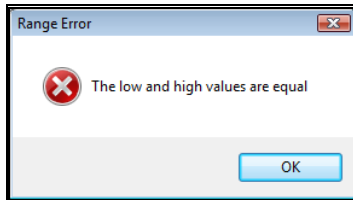
Calibrating Outputs

The procedure is similar except that you enter the low value that you want to send to the output into either the **Scaled** or **Raw** field depending on what you selected in the configuration. Click **Set** to send this value to the device and then measure (in some way) what the actual effect is on the output. Enter this measured value into the **Low** field and click **Set**.

Repeat for a high value and then finally click **Calibrate** to apply the changes to the signal database.

Points to note.

- The **Low** and **High** values specified must not be the same.
- The **Raw** value must be different at the **Low** and **High** engineering values. If the values of the Raw or Scaled values are the same the following error message will be displayed.



- Exiting Calibration without saving will cancel any calibration carried out.

Calibrating Multiple Inputs and Outputs

If there are lots of signals that are of the same type and they can all have their input values controlled simultaneously then multiple signals can have their scaled values set and calibrated simultaneously. To achieve this click on the first signal that is to be calibrated, then shift click on the last signal that needs to be calibrated. This should make the grid look as it does below with multiple signals selected.

Signal	Description
1TC01	13: Oven Front Temp
1TC02	13: Oven LHS Temp
1TC03	13: Oven Back Temp
1TC04	13: Oven RHS Temp

To set multiple high or low values click on the **Set Multiple Low Values** or **Set Multiple High Values** button.

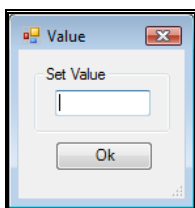


Set Multiple Low Values



Set Multiple High Values

Pressing either button will present you with the following dialog that allows the input of a value that will be set in the appropriate scaled box for all signals that were included in the selection.



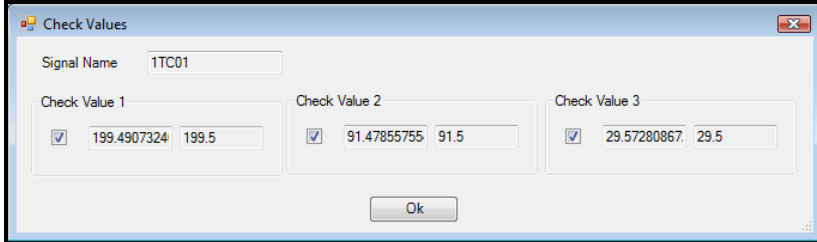
When **OK** is pressed all signals will be given the same scaled value that is related to the Raw value at that instance. The signals can then either be calibrated individually by pressing the appropriate calibration buttons

or they can be calibrated as a group by pressing the **Calibrate Signal Values** button that can be seen below (note all signals that are to be calibrated need to be selected as detailed earlier).

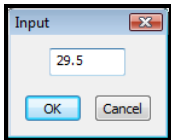


Calibration Check

The **Check** button allows the user to enter check values on any signal currently being calibrated. The following dialog is displayed.



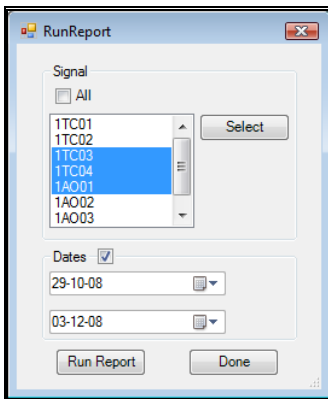
Check values may be entered only for signals that have been calibrated. For an input signal the current value is displayed in three separate fields. Ticking the box beside one of the values holds this value and allows the user to input a manually read value as a check using the following dialog box.



For output signals the user types a value into the edit box. Ticking the box beside it sends the output value to the device. They can then input a check value beside it.

Calibration Reports

Calibration reports can be produced that show any previously saved calibration data. A report can be run by selecting from the menu, **Calibration** followed by **Run Report**. The following dialog will be loaded that will allow you to specify which signals to run a report on **OR** which dates to run a report between.



Reports can be run using either a selection of signals **OR** a set of dates. If a report is required that shows information on all signals that have been calibrated then check **All** button; if a selection of signals from the currently selected configuration are required select them by clicking on the appropriate signals in the list (in the above image signals 1TTC03, 1TTC04 and 1AO01 are selected). If signals that are not in the calibration are required in a report then the **Select** button can be used, this brings up the standard signal selection dialog and multiple signals can be selected by either *Shift* or *Control* clicking multiple signals.

Two dates can be selected by checking the **Dates** box, this overrides any signals that have been selected and provides information on all signals that were calibrated between the dates that are entered in the two drop down boxes. Once criteria have been entered press the **Run Report** button to run a report.

Signal Event Logger

Signal Event Logger

The **Signal Event Logger** logs state changes in **digital** or **finite state** signals. State changes are defined as being changes of value or changes from unknown or disabled to a value, and vice versa. The Event Logger records:

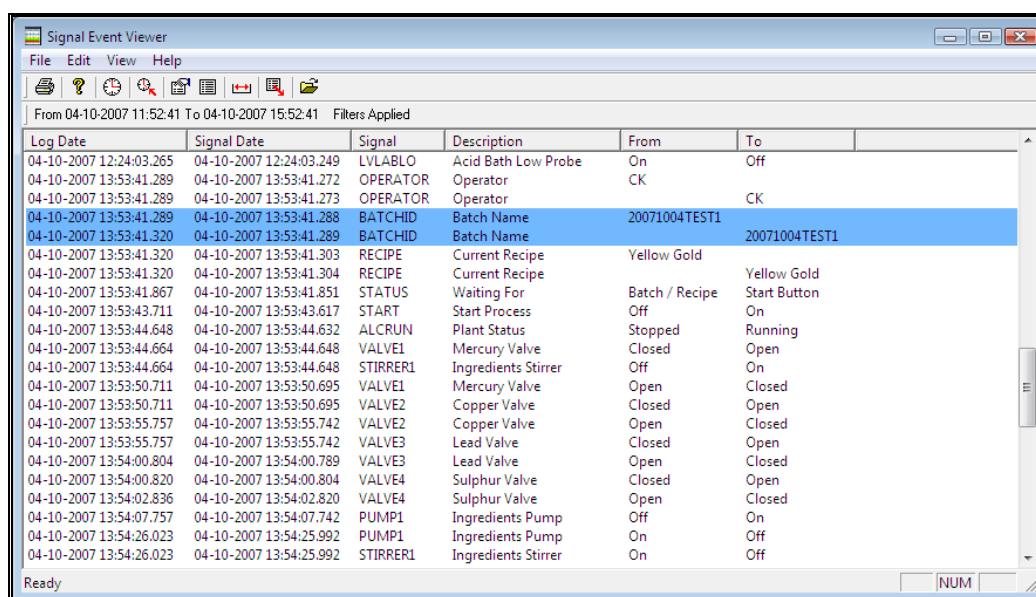
- The name and description of the signal.
- The date and time at which the change occurred, along with the date and time at which it was logged (in some instances these times can differ greatly, depending on such things as the driver being used).
- The state that the signal changed from **and** to. Note that it is not the **value** of the signal that is stored, but the **event text** entry that relates to the signal value at the time of recording. So a pump will go from **stopped** to **running** rather than from **0** to **1**, for example.

Recordings are only made when the signal value changes, and not at any sort of regular interval.

Signal Event Viewer

The Signal Event Viewer is used to view the logged events. It takes the form of a tabular view onto the data and this data can be filtered and sorted as required.

The basic screen is shown below:



The screenshot shows the Signal Event Viewer application window. The window title is "Signal Event Viewer" and it has a menu bar with "File", "Edit", "View", and "Help". Below the menu bar is a toolbar with various icons for file operations, search, and printing. The main area displays a table of logged events with columns for "Log Date", "Signal Date", "Signal", "Description", "From", and "To". The table contains 25 rows of data, with the first row highlighted in blue. The status bar at the bottom shows "Ready" and "NUM".

Log Date	Signal Date	Signal	Description	From	To
04-10-2007 12:24:03.265	04-10-2007 12:24:03.249	LVLBLO	Acid Bath Low Probe	On	Off
04-10-2007 13:53:41.289	04-10-2007 13:53:41.272	OPERATOR	Operator	CK	
04-10-2007 13:53:41.289	04-10-2007 13:53:41.273	OPERATOR	Operator		CK
04-10-2007 13:53:41.289	04-10-2007 13:53:41.288	BATCHID	Batch Name	20071004TEST1	
04-10-2007 13:53:41.320	04-10-2007 13:53:41.289	BATCHID	Batch Name		20071004TEST1
04-10-2007 13:53:41.320	04-10-2007 13:53:41.303	RECIPE	Current Recipe	Yellow Gold	
04-10-2007 13:53:41.320	04-10-2007 13:53:41.304	RECIPE	Current Recipe		Yellow Gold
04-10-2007 13:53:41.867	04-10-2007 13:53:41.851	STATUS	Waiting For	Batch / Recipe	Start Button
04-10-2007 13:53:43.711	04-10-2007 13:53:43.617	START	Start Process	Off	On
04-10-2007 13:53:44.648	04-10-2007 13:53:44.632	ALCRUN	Plant Status	Stopped	Running
04-10-2007 13:53:44.664	04-10-2007 13:53:44.648	VALVE1	Mercury Valve	Closed	Open
04-10-2007 13:53:44.664	04-10-2007 13:53:44.648	STIRRER1	Ingredients Stirrer	Off	On
04-10-2007 13:53:50.711	04-10-2007 13:53:50.695	VALVE1	Mercury Valve	Open	Closed
04-10-2007 13:53:50.711	04-10-2007 13:53:50.695	VALVE2	Copper Valve	Closed	Open
04-10-2007 13:53:55.757	04-10-2007 13:53:55.742	VALVE2	Copper Valve	Open	Closed
04-10-2007 13:53:55.757	04-10-2007 13:53:55.742	VALVE3	Lead Valve	Closed	Open
04-10-2007 13:54:00.804	04-10-2007 13:54:00.789	VALVE3	Lead Valve	Open	Closed
04-10-2007 13:54:00.820	04-10-2007 13:54:00.804	VALVE4	Sulphur Valve	Closed	Open
04-10-2007 13:54:02.836	04-10-2007 13:54:02.820	VALVE4	Sulphur Valve	Open	Closed
04-10-2007 13:54:07.757	04-10-2007 13:54:07.742	PUMP1	Ingredients Pump	Off	On
04-10-2007 13:54:26.023	04-10-2007 13:54:25.992	PUMP1	Ingredients Pump	On	Off
04-10-2007 13:54:26.023	04-10-2007 13:54:25.992	STIRRER1	Ingredients Stirrer	On	Off

The toolbar at the top of the window controls the functionality.



- **Print.**



Prints the currently selected records, taking into account all filtering that is currently in place. The print-out is in a similar collimated form to that shown on the screen, and all selected records are shown, regardless of if there are too many to see on the screen at once (printing can occur across multiple sheets).

- **Help About.**



Displays the Help About box.

- **Select Event Time.**



This button displays a dialog that allows you to select a time around which you want to view events. Events which occurred for a period before and after this time are shown, the time span is governed by a separate button, described later. If a time of 9:00 is selected with a time span of 2 hours, all events from 8:00 to 10:00 will be shown.

- **Re-read The State Log.**

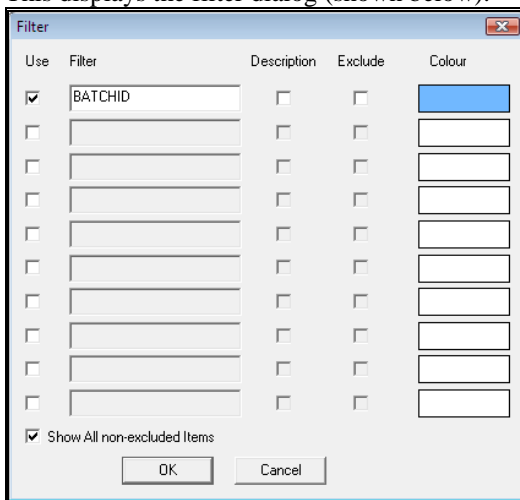


This causes the event viewer to re-read the log. New events may have occurred since the viewer was opened, and this will cause them to be displayed (provided they occurred within the timeframe currently being viewed of course).

- **Filter.**



This displays the filter dialog (shown below).



Filtering is quite simple:

- Check the **Use** box to enable or disable a filter definition.
- Enter the **Filter**, which is simply the name of a signal to be filtered.
- Check or clear the **Exclude** box. If it is checked the signal name entered under **Filter** will not be shown in the Event Viewer. If it is not checked then this signal **will** be shown, but only this signal and all other signals that have filters with no **exclude**.
- Select a **Colour** for the signal (if it is not being excluded). When this signal is displayed on the event viewer its row will be highlighted in this colour.
- To show all other entries rather than just those explicitly mentioned check the **Show all non-selected items** box.. (If this final entry were not set up, our display would show only entries for the **BATCHID** signal. As it is, it shows **all** records, but with **BATCHID** highlighted).

- **Columns.**



This button brings up a dialog that lets you remove some of the columns from the Event Viewer window; they are all displayed as standard.

- **Time Span.**

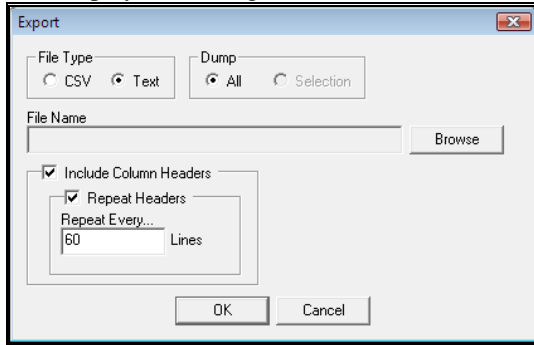


This displays a dialog that allows you to set the time span. The time span is taken as being centred about the currently selected event time (see above).

- **Export.**



This displays the dialog shown below.



- **File Type.**
Data can be exported as either CSV or Text. If it is exported as text an attempt is made to columnate it by padding with spaces, though this does not work perfectly.
- **Dump.**
If **All** is checked then all of the records in the viewer are dumped. If **Selection** is checked then only the **selected** records are dumped (i.e. records that have been clicked on in the list – to select more than one record use the shift or ctrl keys, in the standard windows way).
- **File Name.**
Use the **Browse** button to select the name of the file to dump the data to.
- **Include Column Headers.**
If this box is checked then the first row of output will be the column headers. If the output style that was selected is **Text** then the option to repeat the column headers is given. If selected you must also give the frequency of the repetition (it should equate to the page length on which you expect to print the data, if required).
- **Sorting.**
To sort the data in different ways simply click on the relevant column heading. To reverse the sort order, click the column again. Note that the data can only be sorted by **Log Date**, **Signal Date** or **Signal**. If the data is sorted by anything other than **Log Date** the secondary sort is always on Log Date.

Note that all filters etc that are applied will remain in effect the next time the Event Viewer is run, so on subsequent runs make sure that the filtering is still as you expect it to be.

Black Box Recorder

Overview of Black Box Recorder

If and when a problem occurs it is useful to see actual signal values for a period of time immediately before the problem occurred. The Black Box facility allows the user to do this. If required it runs in the background maintaining a circular buffer of previously selected signals. The user defines the rate at which these signals are recorded and also the time they are recorded for, before being overwritten.

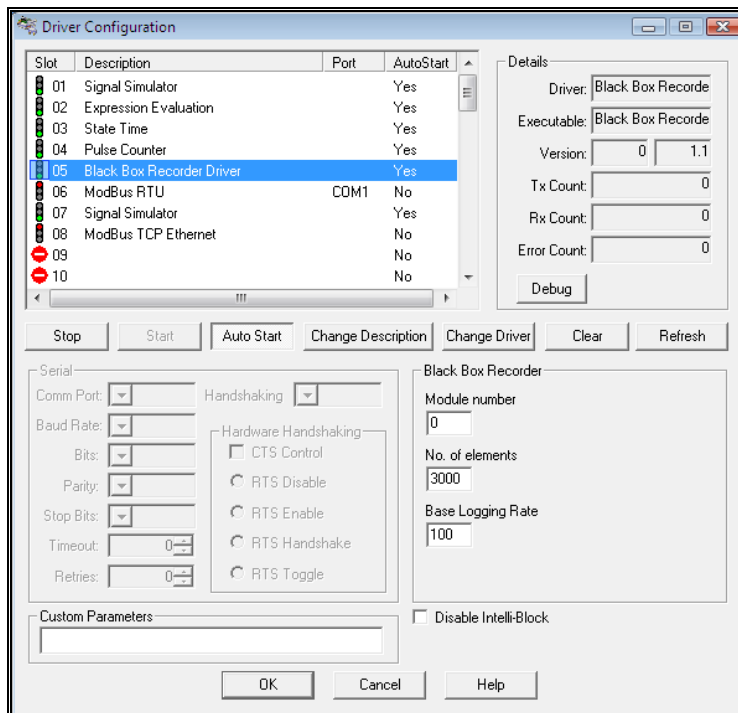
Several Black Box Recorders may be run on one system each looking after up to 200 different signals. At any point in time a dump of these signals may be requested and written to an access database. The Black Box Recorder is run as a driver from the driver wizard and can be told to dump the data or to change the rate at which it logs. SLANG Extensions are available to request a dump or to change the logging rate.

Using Black Box Recording

Setting Up The Driver

In order to use the Black Box Recorder use the Driver Configuration dialog. Select a slot and choose the Black Box Recorder Driver. Three values need to be input and they are shown in the bottom right of the screen.

- **Module Number:**
This can be any number between 0 and 99. No two Black Box Recorders can have the same module number.
- **No. of elements:**
This is the maximum number of each value stored in the circular buffer.
- **Base Logging Rate:**
This is the rate at which the values are recorded in milliseconds. It is possible to change the logging rate dynamically using a SLANG Extension.



More than one Black Box Recorder may be configured, all in separate slots. The reasons for having more than one Black Box Recorder running at any one time, are, if you want to monitor a large number of signals, or if

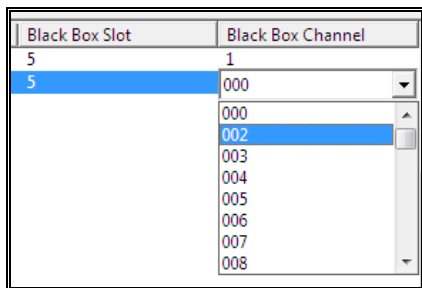
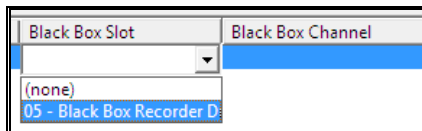
you want to dump only a certain group of signals to an access database. There is a SLANG extension that causes all the signal values being monitored by one Black Box Recorder to be copied to an access Database giving a snapshot of the signals at a particular time. Parameters may be used to specify the size of the snapshot and the interval between signals.

Note that if the Black Box Recorder is stopped using the driver configuration and restarted any drivers used to generate the signals that the Black Box Recorder is recording should also be stopped and started.

Configuring Signals To Be Recorded

Now Black Box Recorder needs to be told which signals are to be recorded. This is done in the Signal Database Configuration.

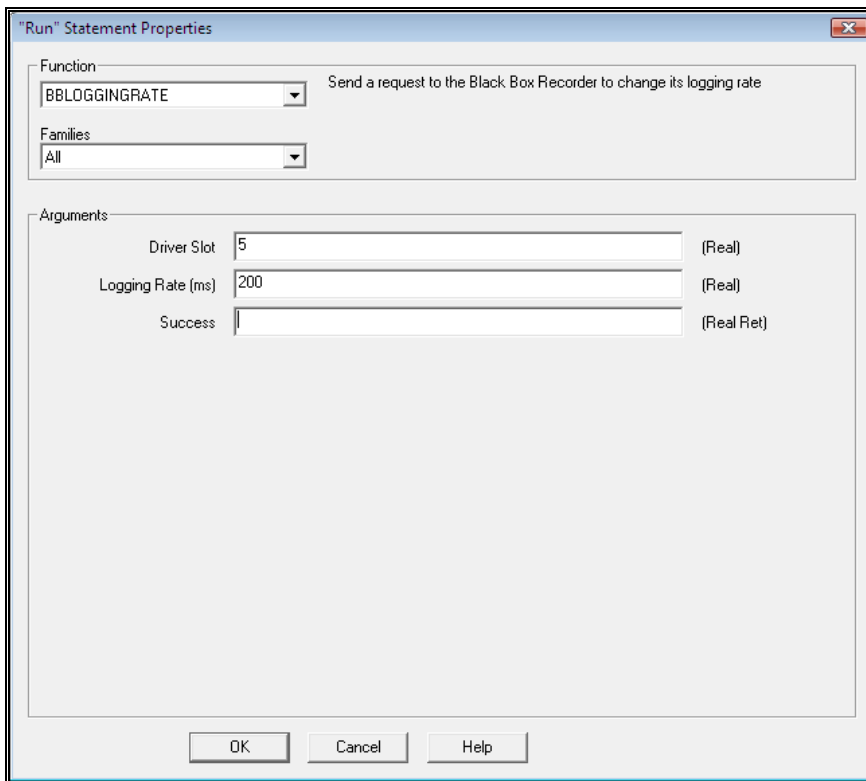
Select the signal that you want to be recorded by a Black Box Recorder and select the Black Box Slot and Black Box Channel numbers from the dropdown list (see Signal Database Configuration for more information).



SLANG Extensions for Black Box Recorder

BLoggingRate

This allows the user to change the logging rate at which the Black Box Recorder records the signal values. The slot number of the Black Box Recorder that the new logging rate will apply to needs to be input as well as the new logging rate in milliseconds. The new logging rate is active immediately. Putting a 0 in the Logging Rate field will set the logging rate back to the base logging rate specified in the Driver Configuration.



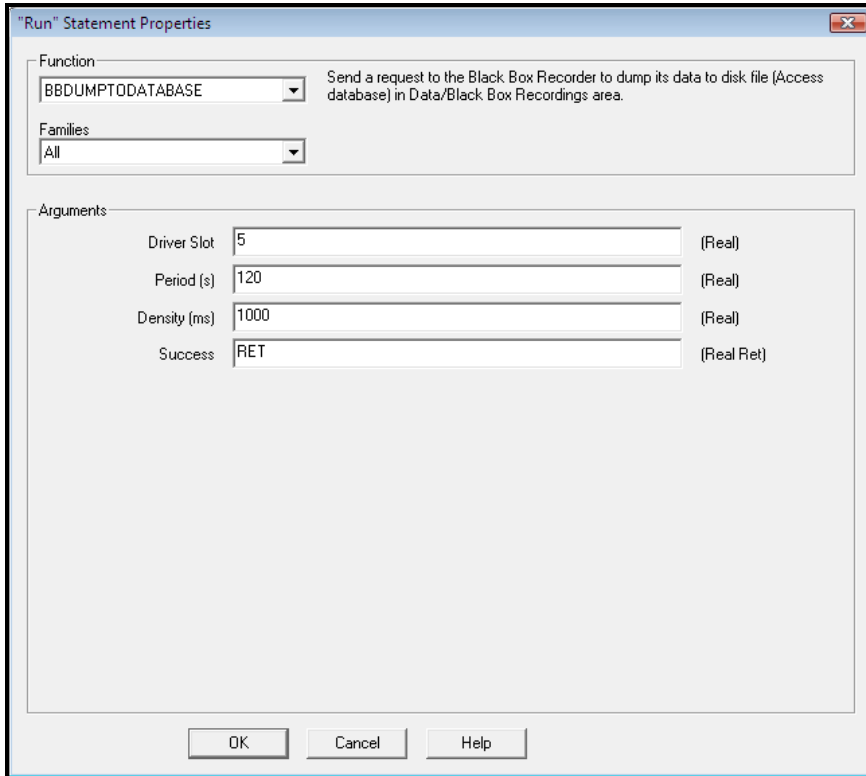
The screenshot shows a dialog box titled "Run" Statement Properties. It contains the following fields and controls:

- Function:** A dropdown menu with "BBLOGGINGRATE" selected. To its right is the text "Send a request to the Black Box Recorder to change its logging rate".
- Families:** A dropdown menu with "All" selected.
- Arguments:** A section containing three input fields:
 - Driver Slot:** A text box containing the value "5", with "(Real)" to its right.
 - Logging Rate (ms):** A text box containing the value "200", with "(Real)" to its right.
 - Success:** An empty text box, with "(Real Ret)" to its right.
- Buttons:** "OK", "Cancel", and "Help" buttons are located at the bottom of the dialog.

BBDumpToDatabase

This allows the user to dump a snapshot of the signals being monitored by the Black Box Recorder to an access database. All the Signals are written to the access database. The slot number of the appropriate Black Box Recorder needs to be input. The period in seconds is the amount of data to be snapshot. The density is the time interval between readings.

In the example below an access database would be created with the last two minutes of data from the signals configured to be using the Black Box Recorder in drivers slot 5. The values will be recorded once a second giving a database with 120 records. If the density was 50 you would still get the last 2 minutes of data but there will be 20 values every second giving a database with 2400 records.



The screenshot shows a dialog box titled "Run" Statement Properties. It contains the following fields and options:

- Function:** A dropdown menu set to "BBDUMPTODATABASE". To its right is the text: "Send a request to the Black Box Recorder to dump its data to disk file (Access database) in Data/Black Box Recordings area."
- Families:** A dropdown menu set to "All".
- Arguments:** A section with four rows of input fields:
 - Driver Slot:** Input field containing "5", with "(Real)" to its right.
 - Period (s):** Input field containing "120", with "(Real)" to its right.
 - Density (ms):** Input field containing "1000", with "(Real)" to its right.
 - Success:** Input field containing "RET", with "(Real Ret)" to its right.

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

What Is The Expert Data System?

The Prodigy Expert Data System provides a means to automatically analyse data to determine the cause of a problem, the solution or indeed whether a problem did in fact occur. The system provides facilities to perform the analysis unaided or to prompt operators to perform certain operations or manual analyses and report their findings back to the Expert Data System. At the end of the analysis the system produce a report of the steps taken in the analysis together with the results of each step, and the suggested course of action. It also allows the operator to come back later and enter notes on the result of that action.

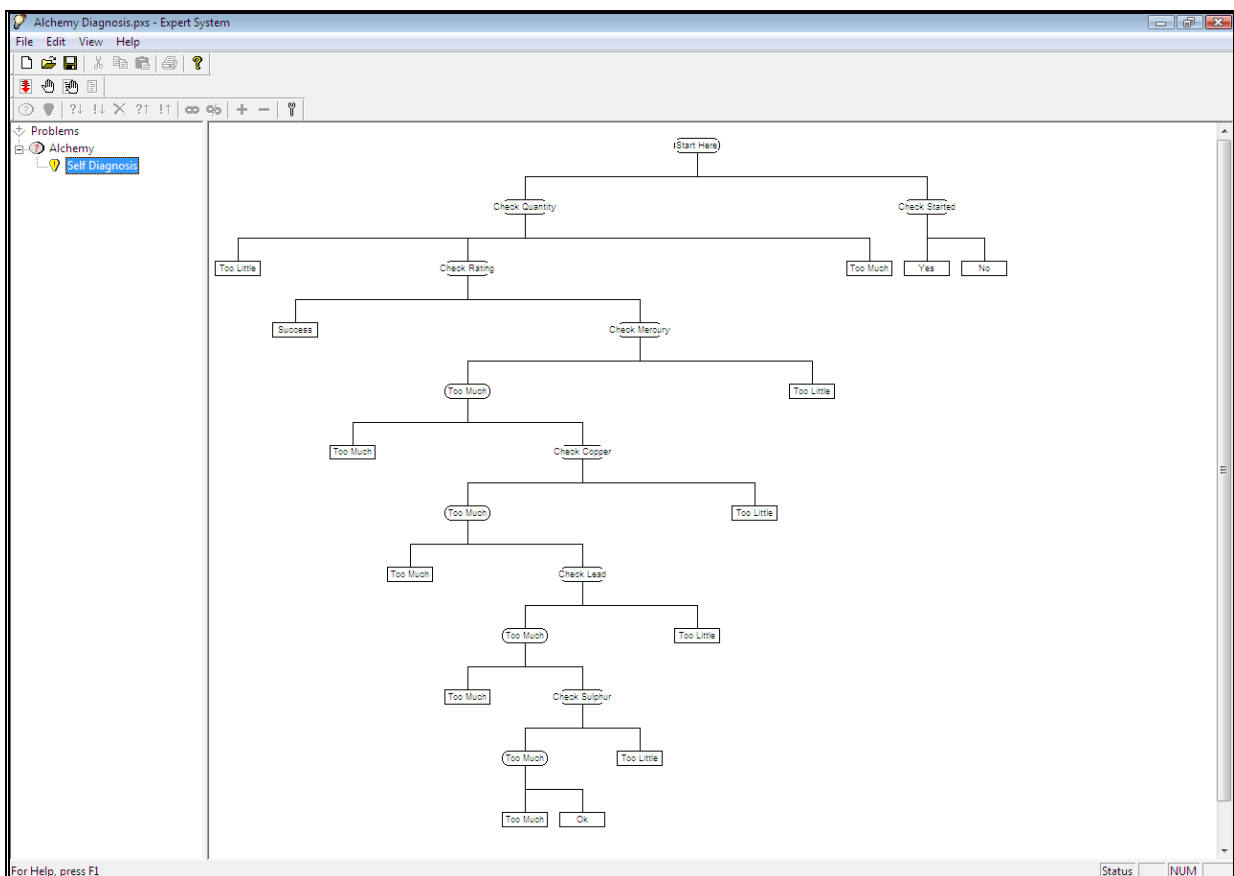
Configuring the Expert Data System

Really, before you can begin to configure the Expert Data System you need to have clearly defined the problem that need to be analysed and the steps that an operator would take in order to investigate and solve that problem. Any data that is required needs to be identified, along with whether or not this data can be collected automatically during the process, or if the operator needs to collect it manually (either during the process or afterwards).

Other parts of Prodigy may need to be configured accordingly so that the right data is collected at the right times, and the course of the decision tree will need to be mapped out in some detail before you begin entry into Prodigy. It is NOT the easiest course of action to sit down and try to build the decisions on-the-fly.

When designing the decision trees remember that at the moment convergence is not allowed - each decision can have only one entry and one or more exits.

The main screen of the Expert Data System is shown below:



On the left is a standard tree control in which you can build up hierarchies of problems or scenarios, the lowest level of which connects to a decision tree, as shown in the right hand pane.

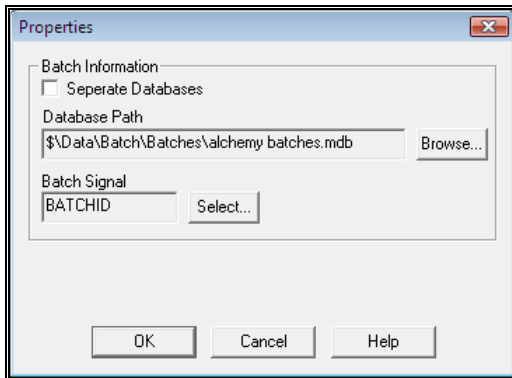
The right hand pane is the most involved and will be described first.

Batch Properties

Click the properties button



The following dialog will appear:



Check the **Separate Databases** box if you are using separate batch databases for this system, or clear it if all batches are in the same database. Then select the **Database Path** by using the **Browse** button.

The **Batch Signal** is the name of the signal that is being used to hold the name of the currently running batch (used by XForm and BDR, and to display the current batch name on Display Builder displays).

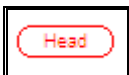
You will only need this information if you are going to use batch variables within your analysis. If you do not specify a Batch Signal or the Batch Signal value is set to 0 at the time an analysis runs, you will be asked to manually select a batch to use for that analysis.

Decision Items

Decision Items decide the route that is taken through the analysis. They can either ask questions of the operator or they can determine the answers to questions themselves. If they are unable to come to a decision automatically based on the information at hand, they can ask the operator for more information or to make the decision manually.

At all stages where operator input is required, sophisticated help can be provided to the operator via words, pictures, formatted documents and even movies. Lets take a look at how you configure Decision Items.

You can load in an existing decision tree or begin a new one. When beginning a new one there will be only one node, a **Decision Item** labelled **Head**.



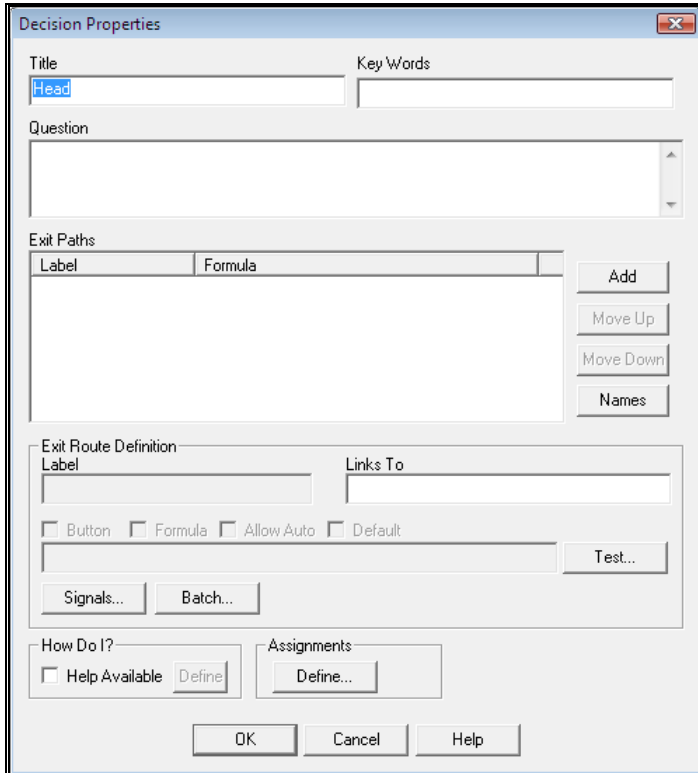
You can change the name of **Head** by double-clicking on the node to display the **Decision Properties** (see later).

Build a tree by adding new **Decision** or **Action** nodes below the current node.

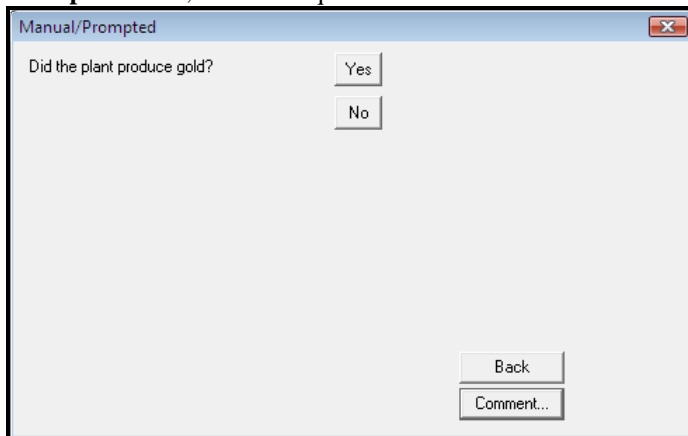
Add a new Decision Item below the current item by clicking the button on the toolbar.



The **Decision Properties** dialog will be displayed, and the new **Exit Path** (between the current node and the newly inserted item) will be highlighted (**Item1** in the list below). Exit Paths are the routes out of a node – decision nodes can have several and action nodes can have only one. Which exit path is taken out of a decision node is determined either by automatic formula evaluation or manually by the operator (or in some cases by a combination of both).



- **Title.**
This is used to label the node on the graphical display.
- **Key Words.**
These are used to aid the finding of nodes that relate to a particular topic. Enter as few or as many as you want.
- **Question.**
This is the string that is displayed to the operator when the analysis is run in either **Manual** or **Prompted** mode, or if the required recorded data is not available (see later).



- **Label.**
Enter some text for the **Label**. This will be used as one of the *answers* to the *question* when the

Prompt dialog appears. So for example, **Yes** and **No** could be the labels of exit paths to the question “Did the plant produce gold?” each leading to a different route through the decision tree.

- **Links To.**
This will be used as the **Title** for the newly entered item (you will see it when you view the properties of that item).
- **Button.**
If you want this option to appear as a button when the prompt dialog is displayed, check the **Button** box. If you do not check this box, the option will not be given on the prompt dialog and the item will just be used for formula evaluation (if specified).
- **Formula.**
If you want to enter a formula for the item check the **Formula** box. The Formula field then becomes available. Enter the formula to be evaluated into this box. You can use signals or batch variables in the formulae – signal names begin with a \$ symbol and batch variables with a ^. So, \$goldrt is a signal and ^goldrt is a batch variable. In order to use batch variables the system must be configured to use batches (see earlier). Expressions can compare batch or signal variables against constant values or against other batch or signal variables. For example:

- \$goldrt >= 10
- ^goldrt >= 10
- \$goldrt >= ^ratreq

The **Signals** and **Batch** buttons enable you to select signals and batch fields to insert into the formula that you are constructing. The **Test** button will parse the expression and test its validity.

When entering batch variables, the variable name given does not have to exist in the batch database. You can knowingly enter a name that does **not** exist. When the system first encounters this variable it will prompt the operator for the value to be used for it. This value may come from some manual measurement or from some automatically obtained measurement that could not be made available to the analysis. Once the value is entered any other formulae that use that variable will use the same value, so it will not be necessary to enter the same values more than once.

Note that batch variable names can contain spaces, but if they do they must be quoted as in this example:

^^weight of gold produced’.

When the analysis is run in automatic mode, the formulae are evaluated in the order that they appear in the list. The first formula that evaluates to true determines the exit path to take. If none of the formulas evaluate to true, and there is no **Default** exit path (see later), the prompt dialog is displayed showing a button for each item that has the **Button** box checked. The operator then has to make a manual decision.

- **Allow Auto.**
This box should be checked if a formula is to be available for automatic calculation. There is not much point in having a formula that does not allow automatic evaluation, but the option is there if you want to use it.
- **Default.**
This box should be checked to determine which exit path to take if none of the formulae match true. This exit path can itself have a formula, but it does not have to have one. For example, consider the three exit paths:
 - Greater: $A > B$
 - Less: $A < B$
 - EqualThe statement labelled Equal could be given the formula $A = B$, but it does not need to if it is checked as the Default action. Similarly
 - 1: $A = 1$
 - 2: $A = 2$
 - 3: $A = 3$
 - None of the above:

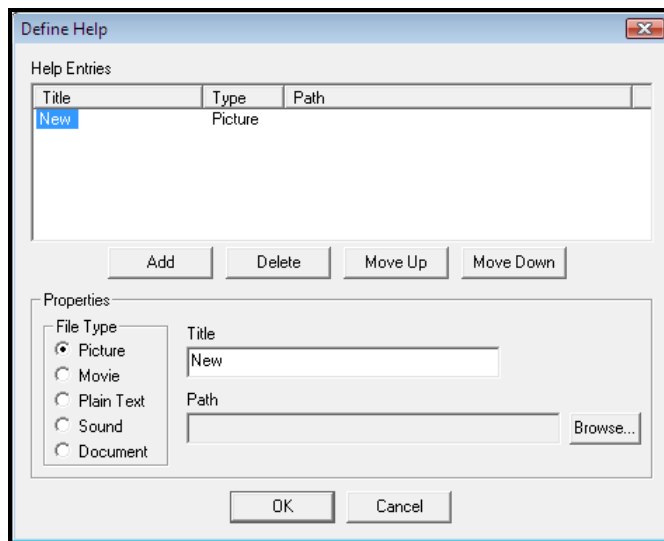
The statement “None of the above” will match any value of A other than 1, 2 or 3. It could have the formula $A > 3$, but if we then later added

- 4: $A = 4$

we would also have to alter the default statement to $A > 4$. Having a default simply offers a catch-all where none of the other statements match. Of course, it may not be sensible to have a default, in which case leave the box unchecked and the system will prompt for the exit to take if no formulae match.

- **How Do I.**

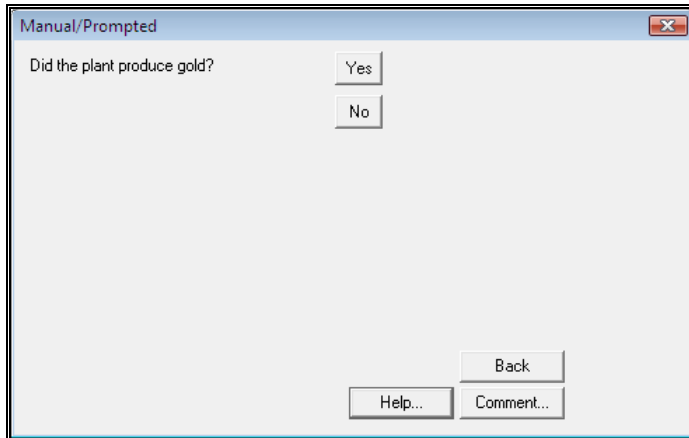
This allows you to specify help for a question. Check the box and then click the **Define** button. The following dialog appears:




Click the **Add** button to insert a new item. Click on this item in the list and then give it a **Title**. Select the **File Type** and then choose a file of the relevant type by using the **Browse** button. File types are:

- **Picture.**
This can be a .bmp, .jpg or .pcx file.
- **Movie.**
This can be a .mpg or .avi file.
- **Plain Text.**
This is a .txt file (contains no formatting etc).
- **Sound.**
This can be a .wav file.
- **Document.**
This can be a Word™ documents (.doc), Acrobat™ documents (.pdf) or plain text documents (.txt).

When the prompt dialog is displayed during runtime, if help has been defined the **Help** button will be available




If you click **Help** the list of help items that you entered is displayed. Select one from the list and press **Show** to display the help in the relevant type of window (note that .pdf and .doc files rely on having Word™ or Acrobat™ Viewer installed).

You can also insert decision items *above* the current item by clicking the insert above button: 

Action Items

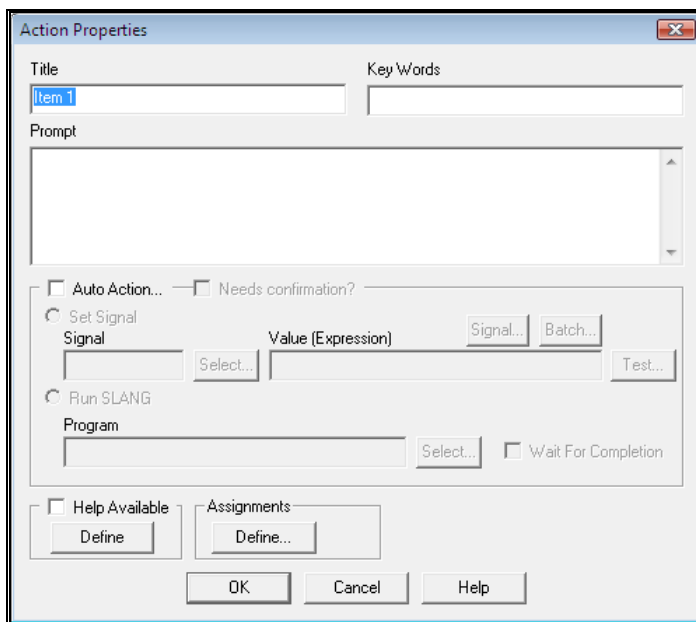
Action items can either ask the operator to do something (and provide them with detailed instructions on how to do it) or they can perform actions automatically by setting signal values or running SLANG programs.

Click on an existing node and then click the button on the toolbar to add action Items: 

If the existing node was an action item the Action Properties dialog for the newly created item will be displayed.

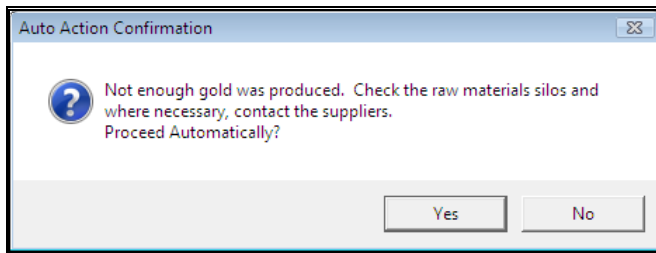
If the existing node was a decision item you will have to provide exit path details for the path to the new action item (the Decision Properties dialog will be displayed with the new exit path highlighted). Complete the relevant details and exit the Decision Properties. Then right click on the newly created action item.

You should now see the Action Properties dialog.



- **Title & Keywords.**
These fields are as for decision items.

- **Prompt.**
This is what the operator will see when this action node is arrived at. This could just provide information to the operator or it could issue them with instructions.
- **Auto Action.**
To automatically perform an action, check the **Auto Action** box. The fields inside this grouping will become active.
- **Needs Confirmation.**
This box lets you indicate that the operator should first be asked whether the auto-action should be performed. They will be presented with the prompt dialog along with the question **Proceed Automatically?** and **Yes & No** buttons.



The type of auto-action to be taken is either to set a signal value, or to run a SLANG program.

- If you choose to set a signal value click the **Set Signal** button and then use **Select** to select the signal. Enter the **Value** to set the signal to. This is an expression, and can be a constant value or an evaluation using other signal or batch values. The same rules regarding variable names apply for this expression as for decision expressions.
- If you click the **Run SLANG** button you must select the SLANG program to run by using the **Select** button. The analysis can then be made to **Wait For Completion** which means it will not proceed to the next item until the SLANG program has run to completion. Whether or not you check this box depends on the actions that the SLANG program will perform.
- As with decision items, help can be provided for the operator should they require it by clicking the **Help Available** and **Define** buttons.

Action Items can only have one exit path, and this can be either another action or a decision.

You can also insert decision items *above* the current item by clicking the insert above button:





Deleting

You can delete items by clicking on them and then pressing the **Del** key on the keyboard or the **Delete** button on the toolbar.



Running An Analysis

There are three modes of running

-  **Manual.**
The operator is asked each question and has to decide manually which answer to choose. Full help is available, as is the ability to enter comments at each step of the analysis, maybe indicating that the decision wasn't clear-cut and saying why they made the choice that they did, and so on.
-  **Prompted.**
This is kind of semi-automatic. The operator is still prompted at each stage, but has the option to accept the computers suggestion (which will be based on any formulae that have been entered).



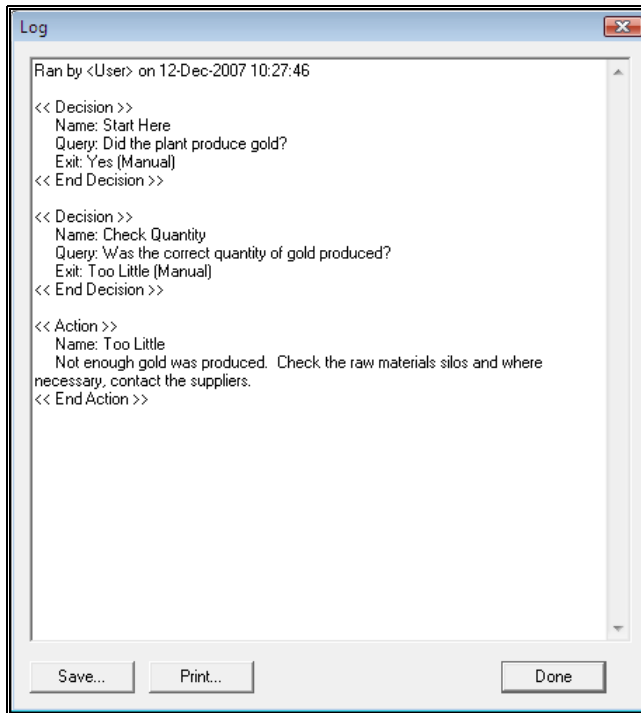
- Automatic.

The computer automatically makes those decisions that it is able to, based on any formulae that have been entered. There can manual decisions within any analysis, in which case the operator will be prompted, and also there may be some decisions for which the automatic formulae fail to come to a solution, and the operator will be asked in these situations too.

After an analysis has run you can view the log by pressing the View Log button:



The log viewer is as follows.



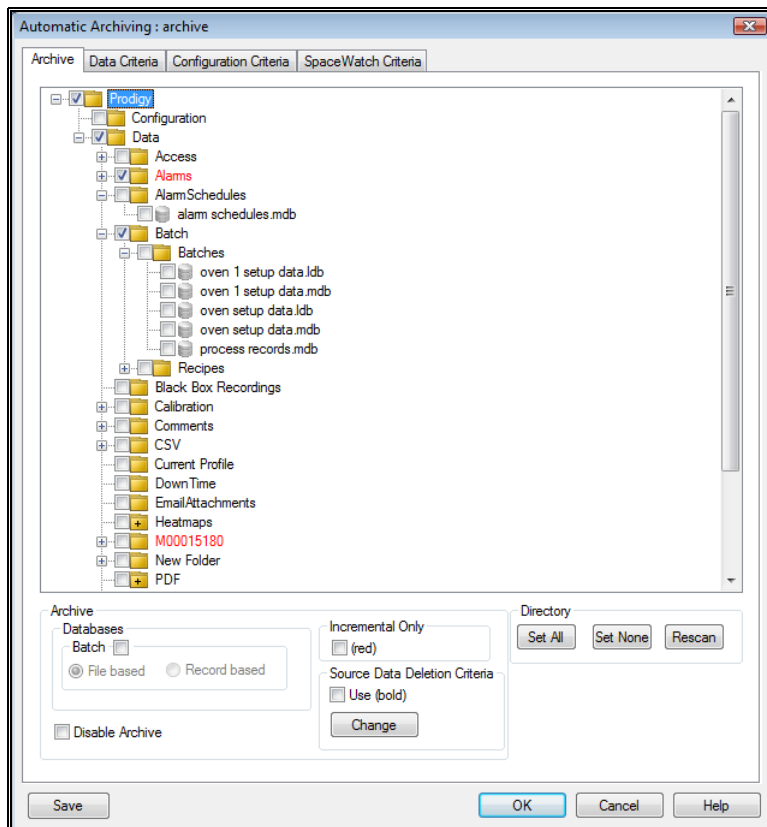
It is in a kind of “meta language” and shows the path taken through the decision tree, with the question that was asked and the exit path that was taken, along with an indication as to whether it was manually or automatically selected. Also when automatic calculations are performed, the values of the variables used are shown too, along with an indication of whether they were obtained automatically (from signal or database) or entered by the operator.

The log file can be saved away by clicking the **Save** button and printed by clicking the **Print** button.

Automatic Archiving

Automatic Archiving

Automatic Archiving handles not only the archiving of old data, but also the backup of current data, the configuration and the management of free disk space for continuously recorded data.



The configuration screen for Automatic Archiving is shown above.

It makes good sense to limit the amount of current data and to move older data to some permanent archive storage, such as tape or CD. One reason for this is that having 1000's of directories degrades performance when viewing the data across a network, so it is better to keep the amount of data folders to a sensible limit.

The archive software works by creating a directory inside the archive area named Data (the data directory might be within some other directories depending on how Automatic Archiving has been configured). Appropriate data that has been selected for archiving is copied into the directory.

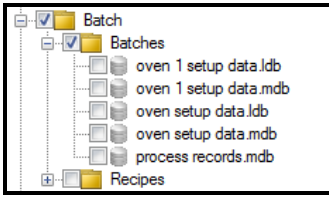
Archive

The **Archive** tab allows the specification of which files and folders will be archived by Automatic Archiving. As a default nothing is set to be archived and the **Disable Archive** button is checked, this prevents Automatic Archiving attempting to perform an archive when it isn't configured correctly.

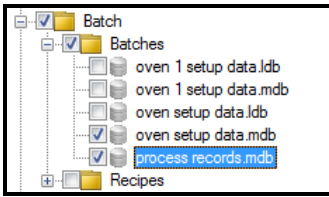
The main area in the archive tab has a tree structure that shows the configuration directory and the structure of the data area. To configure Automatic Archiving to make an archive of the configuration simply check the box next to the configuration folder.

Data can be backed up in a much more specific way that is best described through a few examples using the folders that can be seen in the above window. To configure Automatic Archiving to back up all the relevant data in the data folder simply click the Data Folder, this will cause all the check boxes next to the folders to be highlighted (as above).

If only specific folders or files need to be backed up then the appropriate folders can be selected using the tree structure. Below are two examples of how files in the folder Batches can be backed up both with different results.



In the above example the whole of the batches folder including the 5 files that can be seen will be backed up to the specified archive location. The Recipes folder will not be backed up. If the Batches folder was not selected and the furthest point down the tree that had been selected was the Batch folder the both Batches and Recipes would be backed up including all files located within the folders.



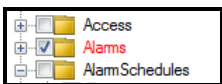
In this example only the files 'Oven Setup Data.mdb' and 'Process Records.mdb' would be backed up. If one of these databases was actually a folder then any files that were located in the folder would be backed up.

Databases

This section is used to configure the archiving of batch databases; it indicates the type of batch database to the Automatic Archiving program so that it can handle the databases in the appropriate manor. If batches are configured to be **File Based** then the folder that contains all the database files should have the **File Based** radio button checked. If batches are configured to be **Record Based** then individual database files can be selected or whole folders. If a folder is selected then the property that databases are record based will filter to all databases within that folder.

Incremental Only

If **Incremental Only** is selected then the text relating to the particular folder or file that the option is set for will turn red in the main window as can be seen for Alarms folder in the image below.

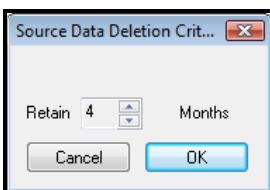


If **Incremental Only** is set then files will only be archived if the file has been updated or changed since the last time an archive was done. This can avoid huge amounts of data having to be copied every time an archive routine is performed; the reduction in data that needs to be transferred reduces the strain that is placed on a network.

Note: **Incremental Only** is always set for the Alarms and M00015180 directories due to the nature of the data that they store.

Source Data Deletion Criteria

This option will cause files to be deleted from the main system if they are over a certain age that is specified by pressing the change button. The dialog to change the time can be seen below.



If the option is set to delete files of a certain age then they will appear in a bold font in the tree structure. If the file that is to be deleted is a batch database then records within the database that are older than the specified time period will be removed from the database.

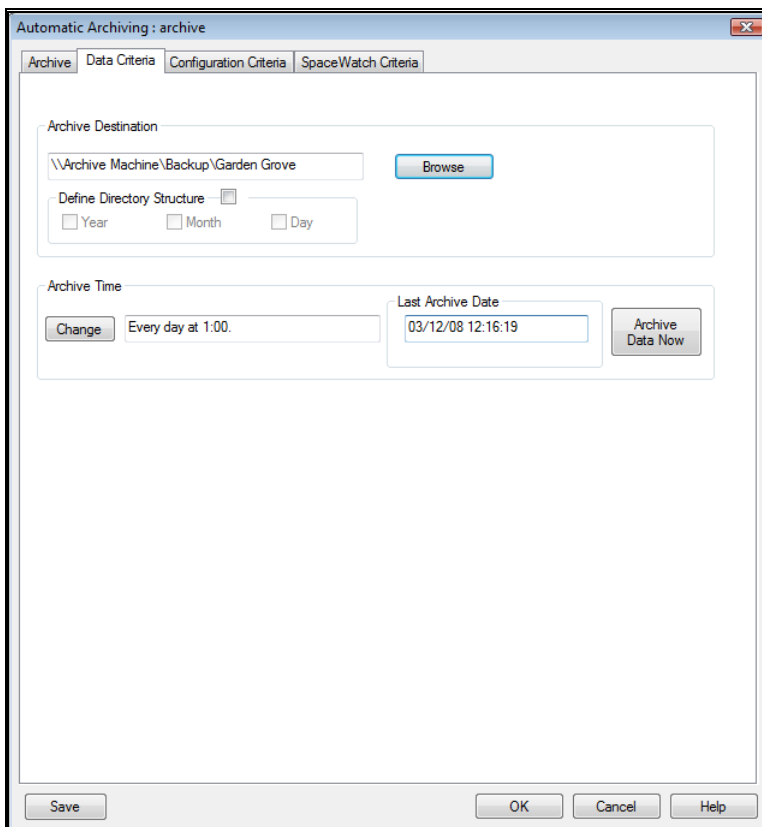
Note: the amount of time files are kept for (in months) is the same for all files in the system that have this option set.

Directory

This area contains three buttons; **Set All**, **Set None** and **Rescan**. If you select a part of the tree structure and press **Set All** this will set the tick boxes for everything one level below that item, pressing **Set None** will do the reverse and remove anything that is set. The **Rescan** button will search the tree for new files or folders that have been created and then add them to the directory structure in archiving configuration.

Data Criteria

The **Data Criteria** tab allows the location, the frequency and the directory structure of the data that will be archived to be configured.



Once files have been archived no attempt is made to remove files from the backup area that have subsequently been deleted from the live area – the backup is purely incremental. For this reason you may need to manually trim the backup area from time to time. You can do this by completely deleting it (once you are sure a hard-copy exists somewhere)

Archive Destination

This specifies the path of the directory that is used for storage of archived data. Use the **Browse** button to specify the backup path. This is most likely some network drive or disk area that is part of a regular off-disk backup procedure. The sub-directories **Data** and **Configuration** are made within this area.

Define Directory Structure

As a default none of the check boxes are selected and data will be stored in one folder that is named data and is placed in the *archive destination* folder. The options here can be used to define a directory structure for

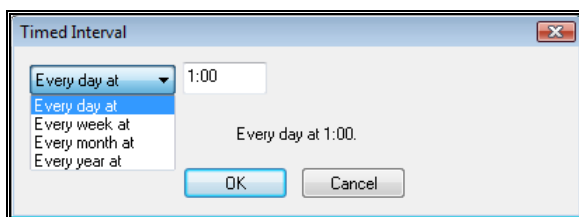
where the data will be stored. Checking the **Define Directory Structure** box will also cause **Year** to be selected, in this case when an archive is performed a directory will be created that uses the date when the archive is being performed to find what year it is and this will be used as the name of the directory where the data folder will be placed. The three directory structures that would be created if an archive was performed on the 25th December 2008 are shown below depending on how many of the options are selected.

- **Year**
Archive Directory\2008\Data
- **Month**
Archive Directory\2008\Dec\Data
- **Day**
Archive Directory\2008\Dec\25\Data

If in 2008 **Year** is selected and an archive is performed a data directory will be created within a folder named 2008. Any archives that are performed after the folder has been created in 2008 will just update the data folder within the 2008 folder with any new information that is needs to be backed up. When the first archive is performed in 2009 a new folder named 2009 will be created where any data that has not been backed up since the last archive will be stored. The same applies for the month, and day settings however it must be noted that a sensible directory structure (if any) needs selecting that will complement the archive time (see below).

Archive Time

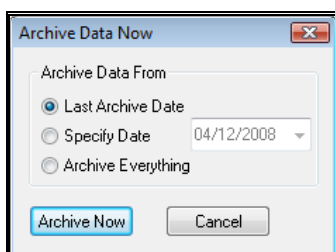
Pressing the **Change** button will bring up the following **Timed Interval** dialog; this allows the selection of the time interval when backups of the data will be taken:



Choices are **Every Day** at a given time, **Every Week** at a given time on a given day; **Every Month** at a given time on a given day of the month; **Every Year** at a given time on a given day of a given month. The backup will begin at the time specified here.

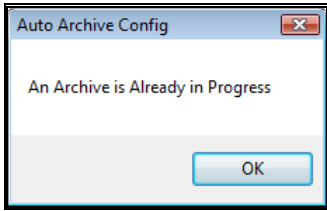
The **Last Archive Date** shows you that last date that files were backed up. When a backup is run, this date is automatically set to the date and time of the last backup so that all files created or modified after that date will be backed up.

The **Archive Data Now** button will bring up the following dialog that can be used to cause prodigy to perform an archive.



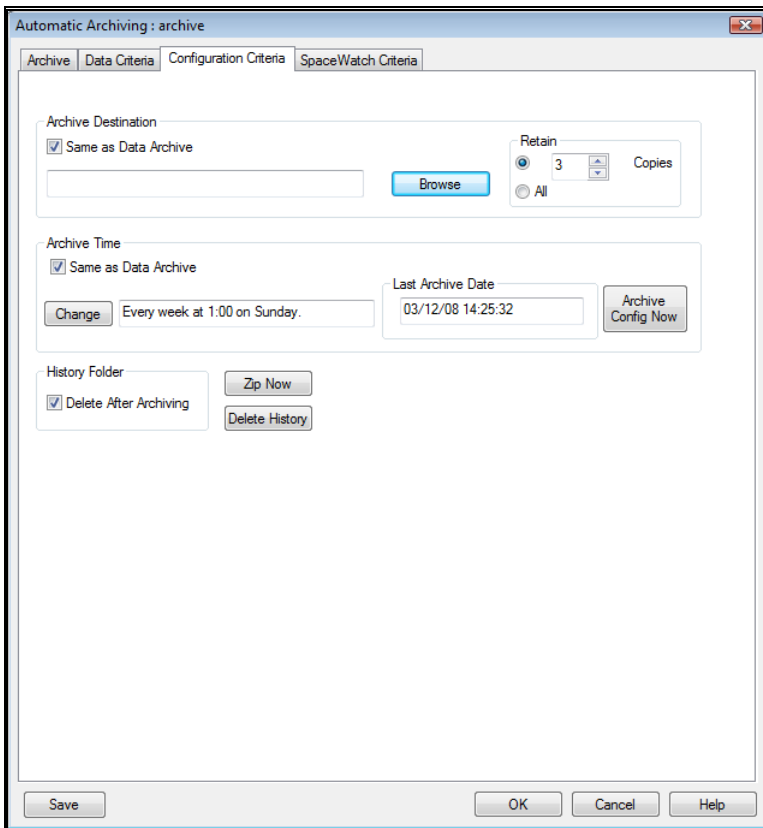
There are three options available Last Archive Date, Specify Date and Archive Everything. If **Last Archive Date** is selected then the archive program will archive any files that have been modified since the **Last Archive Date** (described earlier). If **Specify Date** is selected then a date can be selected and any files that have been created or modified since the date that was selected will be archived to the specified location. Finally if **Archive Everything** is selected then all data that has been selected on the archive tab, for archiving will be archived to the specified location.

If an archive is being performed and the **Archive Data Now** button is pressed the following dialog will be displayed that will indicate an archive is already in progress; the dialog to perform an archive will not be available until the current archive has finished.



Configuration Criteria

To create backups of the configuration first make sure that the configuration folder is selected on the archive tab, then the **Configuration Criteria** tab can be set up.



The **Configuration Criteria** tab is similar to the **Data Criteria** tab just there are some extra option available.

Archive Destination

This specifies the path of the directory that is used for storage of archived data. Use the **Browse** button to specify the backup path. This is most likely some network drive or disk area that is part of a regular off-disk backup procedure. The sub-directories **Data** and **Configuration** are made within this area. If the archive location is the same as the Data Archive location then checking the **Same as Data Archive** box will cause the system to just use the information that's specified on the **Data Criteria** tab.

Retain

This allows the number of copies of the configuration that are retained at the archive destination to be specified. If **All** is selected then a copy will be taken every time an archive is performed and none will ever be deleted, if a number is specified then when that number of configurations has been backed up archive will start deleting the oldest configuration every time a new one is taken. To delete the older configurations from the archive destinations you must have the ability to edit files in that area.

Archive Time

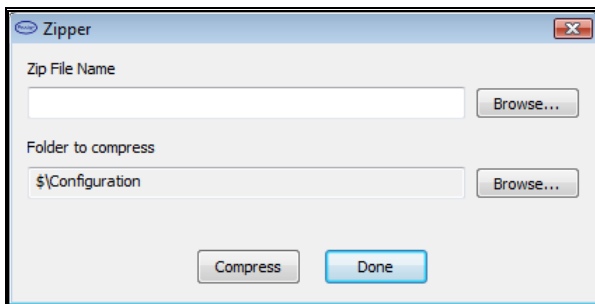
The options in this section are as described for the same section of the **Data Criteria** tab. There is the additional feature of the **Same as Data Archive** box will cause the system to just use the information that's specified on the **Data Criteria** tab.

When the **Archive Config Now** button is pressed an archive of the configuration is stored in the Archive Destination, the configuration folder has the date attached to the end of it so that different configurations are identifiable. A configuration taken on the 25th of December 2008 would have a file name of Configuration.20081225.

History Folder

If the **Delete After Archiving** check box is selected then when an archive of a configuration is performed any history that is stored about modifications to the signal database will be deleted from the disk and removed from the system.

Pressing the **Zip Now** button will load up the following dialog:

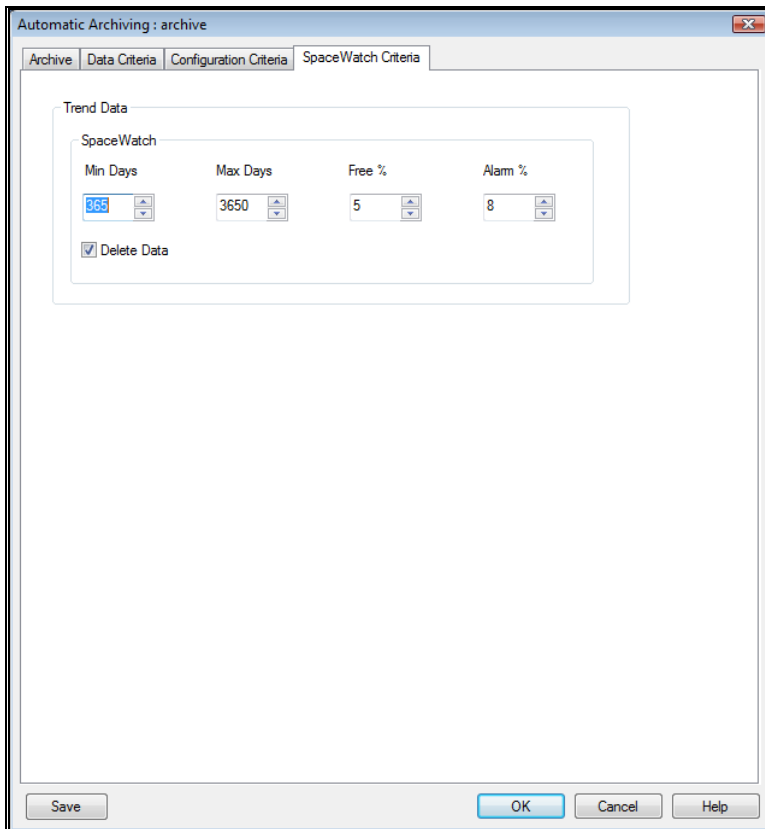


This provides a facility to compress the Prodigy configuration directory; to use the Prodigy Zipper simply click the browse button next to the **Zip File Name** edit box. Then enter a name for the configuration and select a location for the file to be stored; pressing the compress button will create a zipped file that can be accessed by appropriate zipping utilities, Windows XP or Vista. Prodigy Zipper can also be used to zip other files or folders by pressing the lower of the two browse buttons and selecting an appropriate file.

The **Delete History** button will delete the signal database history that is stored on the current system; if a system is fully working then there is probably no reason to need to keep any history of the signal database and changes made to it during the development of the system. Before the signal database history is deleted confirmation that this action is required will be requested via a pop up dialog.

SpaceWatch Criteria

Spacewatch is responsible for ensuring that there is sufficient disk space available for the continuous data recording. It does this by monitoring the % of disk space that is free.



When the free disk space falls below the **Free %** threshold that has been set (5% in the example above) it will begin to delete old continuous data.

The amount of data that will be deleted varies according to the **Min Days** and **Max Days** parameters above. In this example, data will be deleted until there is no more than 10 years worth of data (3650 days). If the free space threshold is still exceeded data will continue to be deleted, but deletion will stop at 30 days (i.e. the system will always keep the last 30 days of data).

If the amount of free space falls to the **Alarm %** (8% in our example) an alarm will occur, which will be seen on the **Alarm Display** and also in **Alarm Reports**.

If your historic continuously recorded data is important to you, you should ensure that it is archived away using the **Archive Of Old Data** facility described earlier.


HTML Reports and Web Upload

Introduction

It is sometimes required to generate a report in HTML format. This may be done in various ways within Prodigy. In addition, these reports can be automatically uploaded to a web server.

Producing HTML Reports

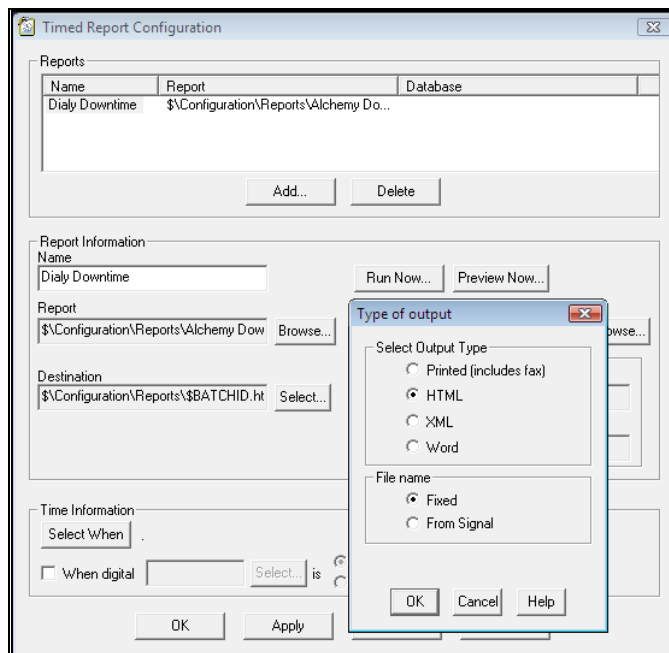
While viewing a report on the screen

When a report is being viewed on the screen it can be exported to HTML format using the **Export** button .

In the dialog brought up by pressing this button, specify **HTML** as the format from the list and **Disk File** as the destination. An image of the report will be generated as an HTML file. If there are any pictures on the report they will be saved as separate files.

Automatically Using the Timed Reports Facility.

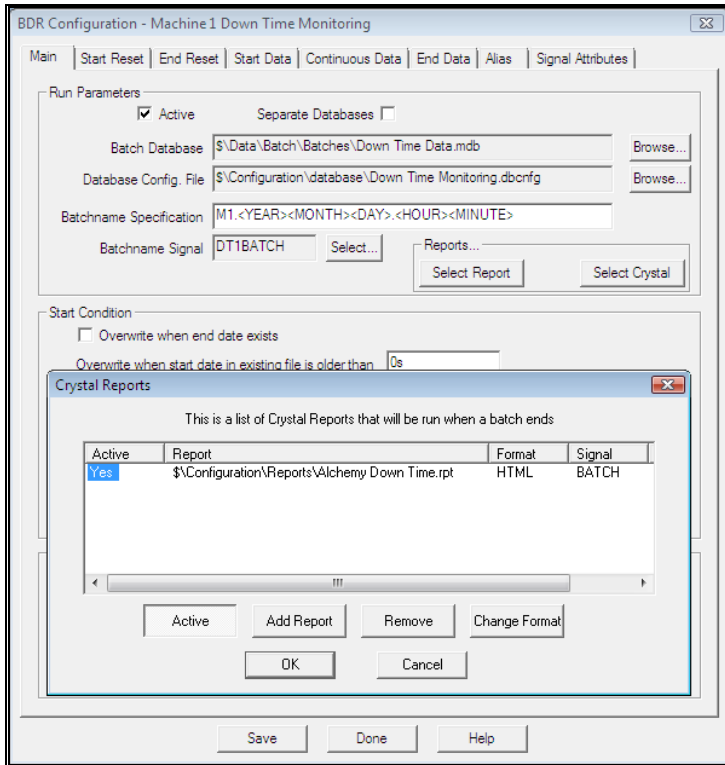
The **Timed Reports Configuration** allows the user to specify a report to be run at a particular time or when a chosen signal is set. This report may be put to a file in HTML format. The user specifies the report to be run and the database. They are asked for the format of the report when they press the button labelled **Select...** next to the **Destination** field.



If they specify HTML as the report format when the report is generated it may be saved to a file which has a fixed name. I.e. if the file exists prior to a report being generated it will be over written with the next report. If this is not acceptable the filename may be taken from the event text of a signal. The event text used is that of the current value of the signal. The event text may be specified when the signal is created by **Signal Database Configuration** or it may set by an **XForm** or a **SLANG** program.

Automatically Generating a Crystal Report at the end of a batch

A Crystal report may be generated at the end of a batch in a similar way to a single page *Proof of Process* report. When configuring Batch Data Recorder, on the Main tab of the Properties dialog press the button labelled *Select Crystal* in the *Reports* group.



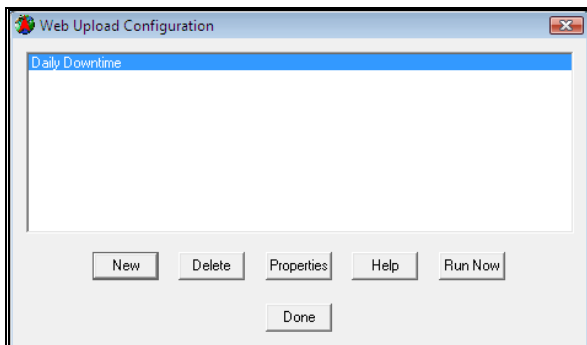
Reports may be added which will run when the batch completes. As with the timed reports the format may be specified to send the report automatically to either the printer or a file in HTML format. The filename is read from the event text of a signal. If this file already exists it will be overwritten.

Uploading HTML files to a Website

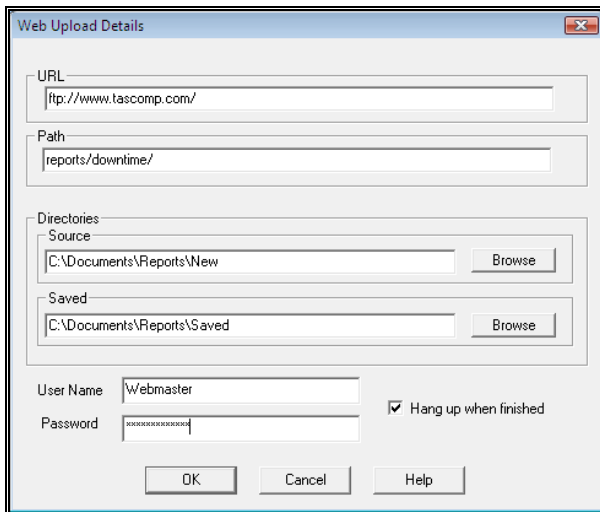
There are two programs called *WebUpload* and *WebUploadConfig* that take care of automatically uploading HTML files to a web site.

Configuring the Details for Uploading Files

The Web Upload Configuration program is similar to other configuration programs in that it saves configuration information in a file. You may configure multiple files.



Pressing the *Properties* button brings up the following screen.



The URL and Path between them give the FTP site address where the information is to be uploaded to. The URL is case sensitive and must be specified exactly with the correct case and the slashes in the correct places. The user name and password are needed to log on to the site. The user must have the correct permission to upload to the specified ftp address. Any other information needed such as the phone number of the ISP is retrieved from the default dial up properties configured on the machine.

The source directory is the path of the directory that is to be monitored for files appearing in it. When one or more files appear in this directory they will be copied to the web address specified by the URL. The name of the file on the web will be the same as the one in this directory. Note, filenames must *not* contain any spaces as this will cause the uploading process to fail. Once the file has been successfully copied to the web it may be copied to a saved directory before being deleted. If the saved directory field is left blank the file is deleted after a successful upload and not copied to the saved directory.

Pressing the *Run Now* button causes WebUpload to run immediately and try to upload any files that exist in the Source directory. This is useful if the WebUpload program has failed to upload some files for some reason (problems with the ISP for instance). Otherwise WebUpload would wait until another new file appears in the Source directory.

Uploading Files

The program *WebUpload* runs continuously. The *Display Launcher* Program starts it. On the *Display Launcher Configuration* the box labelled *Run Web Upload* must be ticked and the configuration file to use must be specified.

Once the program is running it monitors the source directory. When one or more files appear in the directory the program tries to connect to the Internet if a connection is not already there. It uses the information given in the configuration file along with Internet information already on the machine. The files are uploaded, then if no errors occurred and it is required, it is copied to a “saved” directory. The files are then deleted from the source directory. If a file already exists it will be overwritten.

Dual Computer Support

What is Dual Computer Support?

Prodigy can be configured to work in a mode where two computers run the same system simultaneously. Both computers should be connected to the same plant I/O so that they collect their data from the same source but independent of each other.

A Remote Status Monitor runs on each computer and monitors the status of the Prodigy system on the other computer. If anything goes wrong with one of the prodigy systems, or if the computer itself should fail, the Remote Status Monitor logs the fact that a fault has occurred and may also trigger an alarm.

When Prodigy is started on either of the computers, it gives the option of first copying all of the configuration and data files from the other computer onto this computer. This facility allows the computers to be resynchronised after one of them has been down for a period of time.

Setting up Dual Computer Support

Preparing Your PCs

The first thing to do is to make sure that the correct **user rights** are set up. The Prodigy user must be given the rights to “Act as part of the operating system”. This is because the dual support facilities need to log themselves in as a specified user. For details on how to do this, see elsewhere in the manual under **TCP/IP Quick Start**.

Next you must ensure that TCP/IP is installed and running on the computers (again, see **TCP/IP Quick Start**).

Next share the path to the Prodigy directory on both computers. The Prodigy directory is normally at D:\Program Files\Tascomp Ltd\Prodigy, though the drive letter may vary.

Next, make a mapped drive on each computer (call it drive Z for simplicity) that maps to the shared Prodigy directory on the other machine. Make sure that the mapping is set to be reconnected at logon.

Make sure that the *User ID* and *Password* specified in the *Remote Access* section of the Services tab of the Prodigy Control Panel is set to something that is valid on both computers (this is standard procedure when configuring any Prodigy system that communicates between two or more PCs).

Configure the Prodigy System Services to be manual start rather than automatic start. (Do this via the Services icon from the NT Control Panel).

Configuring Dual Support

In order to configure dual support, run the **Prodigy Control Panel** and select the *Dual Computer Support* tab.

How Best To Approach Dual Support

- Install Prodigy on both computers.
- Configure the services to be manual start rather than automatic start.
- Configure the two PCs to have TCP/IP and the correct user rights.
- Share the Prodigy directories on both computers.
- Map the shared directory from the other computer onto a drive on this computer (do this on each computer).
- Use the Prodigy Control Panel to set up the Dual Support Parameters. Do this on one computer and then use the **Send Config** button to send it to the other computer.
- Start Prodigy on the computer that has not been used for development and when prompted, copy the required configuration and data from the other computer.

- Start Prodigy on the computer that has the correct configuration (the one that has been used for development of the system).

Configuring Options

See the relevant section of the manual under *Services*.

When A Fault Occurs

If a fault condition occurs on one of the computers, Remote Status Monitor will report this to the other computer. Depending on how the system has been configured this can then report the fault. Faults that can occur include

- Communication failure – the network has a problem and the Remote Status Monitor cannot communicate with the other computer.
- Remote Status Monitor has failed – it may have died or it may have been deliberately stopped.
- Prodigy has stopped, or is in the process of shutting down or starting up – if any of these conditions exists, the Remote Status Monitor will report it.
- A part of Prodigy has unexpectedly stopped – any unexpected termination of any of the Prodigy sub-system will result in a report by the Remote Status Monitor. This does not include failures of any foreground (display) applications.

When any of these faults occurs, you should check the cause. In the case of a comms failure, first check that Remote Status Monitor is still running on the other computer (use the Control Panel to check this). If it is, then check that the network is OK (use a browser window to try to browse to the other computer). If everything seems to be OK, use the Prodigy Control Panel on each machine to stop and start the Remote Status Monitors. Hopefully that will clear the fault.

If Remote Status Monitor reports that a fault has occurred with a part of Prodigy, it is best to restart Prodigy on the computer that has the fault. That computer should also be showing a dialog that indicates what the fault is, and there should be an entry in the system event log. Depending on what has gone wrong, it may or may not be necessary to resynchronise the computers when Prodigy is restarted.

Prodigy Workstation

What Is Prodigy Workstation

Prodigy Workstation allows a second PC to act as a remote terminal to one or more host PCs running full Prodigy systems. It can switch relatively quickly between systems allowing one Prodigy Workstation to be used to monitor the situation on several Prodigy systems plant wide.

How Prodigy Workstation Works

A program called *Workstation Connect* presents the user with a list of host machines that are running Prodigy. When the user selects one of these machines and presses the **Connect...** button:

- The signal database is copied from the host machine.
- This signal database is automatically modified so that all signals use the network driver and so receive their values from the host machine.
- All Prodigy file paths specified in the registry are modified so that they map onto the file paths used on the host machine. This way, all data and configuration files (except those specific to Workstation) are shared between the two machines.
- A small subset of the Prodigy Services are started up.
- A separate Menu Bar is presented for each machine that Workstation can connect to. This allows different levels of access to be provided on each of these different machines.

Limitations

Prodigy licensing allows a variety of access scenarios with Workstations.

- **Workstation** presents a read-write view onto the remote machine. The process may be controlled and/or manipulated from the Workstation - signal values can be set via any of the available means (Displays, XForms etc.).
- **Viewstation** presents a read-only view of the remote machine. No signal values can be modified by whatever means.
- **Chartstation** presents a Trend-only view of the remote machine. Displays containing anything other than Trends will not load.
- Alarm Display can be run on the Workstation but accepting an alarm on the Workstation has no effect other than clearing the alarm from the Workstation's Alarm Display. The alarm must be accepted on the main computer in order for it to be actually "accepted".
- No configuration of the signal database is allowed by any Workstation license. However, XForms, Displays, DTL, BDR, Shop Floor terminal etc. can all be configured from the Workstation provided the master system's license allows it.

Using Prodigy Workstation

Software Installation

The software installation is exactly as for a normal Prodigy system.

Licensing Requirements

In order to use Prodigy Workstation you must have a Workstation license installed on each machine that is to act as a workstation. Prodigy looks for the presence of a Workstation license and if one is **not** found the system will behave as a **normal** Prodigy system. If NO license at all is found the system will start as a normal Prodigy system in **Demo** mode.

The same workstation license can be installed on more than one workstation PC but only one of those PCs will be allowed to connect to the server at any one time. The server checks that only one instance of any given workstation license is connected at any one time and rejects connection attempts by duplicate licenses.

The workstation will have the same functionality in terms of what Prodigy facilities it can access as the server has. In determining what facilities are licensed the workstation accesses the license on the server – the workstation's own license is merely to inform Prodigy that this machine is in fact a workstation.

Security

If you have implemented a secure system using Users, the Workstation will use this too. Workstations access the same Users database as the server (the database lives on the server) and a user will have only those facilities assigned to them, no matter where they accesses the system from.

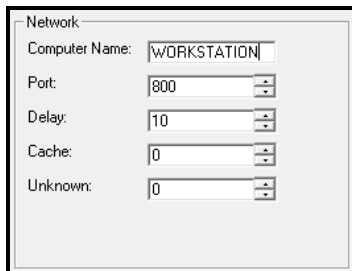
Logging on is per-workstation too. If a user logs on at a workstation they in no way influences who is logged on at any other workstation or on the server.

As mentioned previously, the Menu Bar is on a per-workstation basis too, rather than being global. You can configure each menu bar on each workstation to require certain privilege(s) to access various facilities. In this way a user may be allowed to do some things at some workstations, but not at all workstations.

System Preparation

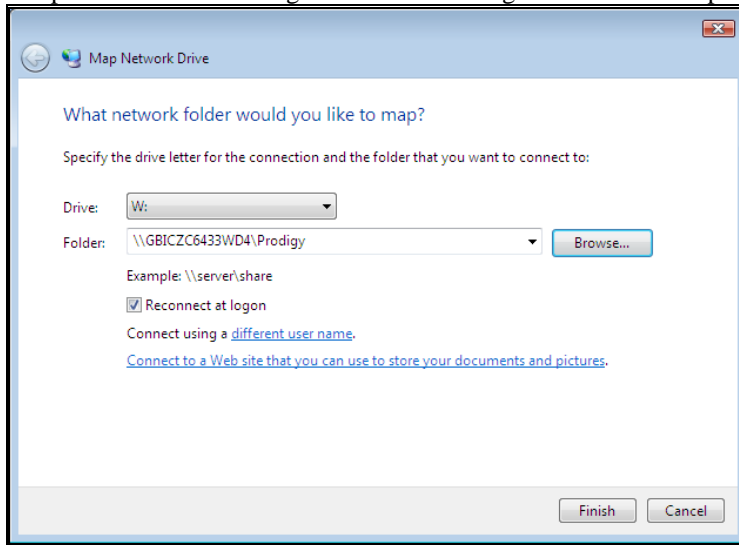
In order to use Prodigy Workstation a little preparation has to be done on the machines that are to be used.

- On the machine that is to run Workstation, configure the Prodigy System Services (services applet from the control panel) to be manual start or disabled (this is not necessary if a **Workstation** installation has been done). If the services are left as auto-start the services will start up each time the Workstation is booted but in an incorrect manner. Though this will result in what appears to be a working connection, the latest copies of the database and other associated files will not have been brought from the host machine, as it is the Workstation Connect program that does this.
- All of the machines must be configured to use the TCP/IP networking protocol (*see the relevant Windows NT documentation or consult your system administrator*). If you are not using Wins/DNS be sure to edit the file `\Winnt\system32\drivers\etc\lmhosts` to contain the names and IP addresses of all of the machines that will be linked.
- The host machine (the one which is running the full Prodigy system) must be configured to run the Prodigy Network Driver.
 - Load the driver into any available empty slot.
 - Enter the name of one of the machines that is to run Prodigy Workstation into the **Computer name** field of the network driver.
 - Set the **port number** to a unique value greater than or equal to 800.



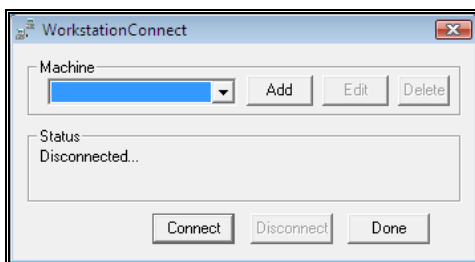
- Be sure to **Start** the driver and also set it to **AutoStart**.
- Repeat the procedure, adding a Network driver for each machine that will be a Workstation, giving each the correct **Computer name** and another unique **port number**.
- See the *Driver Configuration* section of this manual for more information.
- On the host machine the Tascomp directory of the program installation path must be made shareable. This is usually `n:\Program Files\Tascomp Ltd` where `n` is the drive letter. This will allow the Workstation system to access the required license, configuration and data files of the full Prodigy system.

- On the machine that is going to run Workstation you should make a mapped drive to the shared **Tascomp Ltd** folder on the host machine (call this drive W for example) giving any required user-id and password and checking the reconnect at logon box. The example below is for Windows Vista.



Running Prodigy Workstation

Double click on the Workstation Connect icon or select Workstation Connect from the Start button menu. The following dialog will be displayed.



In order to run Prodigy Workstation simply select the required machine from the Machines list and press the **Connect** button. As the connection is made the **Status** message will change to show the current state. When the message reads **Connected** the connection is complete. The menu bar for this connection will be displayed.

To disconnect press the **Disconnect** button. When the status message reads **Disconnected...** the connection has been closed. If you want to connect to a different host you must disconnect from the current connection first.

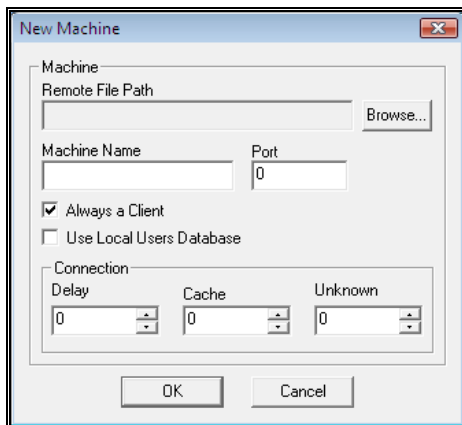
The fields and buttons on the dialog are as follows.

- **Machine.**
This is a drop down list of all machines that can be accessed from Prodigy Workstation. The last machine that was connected to will be displayed as the current selection in the list.
- **Add/Edit...**
This button allows additional new entries to be added to the list of machines that can be accessed via Prodigy Workstation and also allows existing entries to be modified. See later for a more detailed explanation.
- **Delete.**
This button deletes the currently selected machine from the list of those that can be accessed via Prodigy Workstation.
- **Status.**
This shows the current status of Prodigy Workstation. In the dialog above the status is *Disconnected* showing that the services are not running. During the connection phase this message shows the current action being performed.

- **Connect.**
Connects Prodigy Workstation to the machine currently selected in the drop down list and starts the services.
- **Disconnect.**
Disconnects Prodigy Workstation and stops the services.
- **Done.**
Closes the dialog but leaves the current status unchanged.

Adding A New Machine

If the machine you wish to connect to does not appear in the drop down list press the **New...** button to display the following dialog.



This dialog allows you to specify the machine, path and logon parameters for your connection to the host computer.

- **Machine Name.**
The name of the host computer as specified when TCP/IP was configured.
- **Port.**
Enter the port number of the Network2 driver on the host machine with which this machine is to communicate. **N.B.** Port numbers must match.
- **Remote File Path.**
This is the path of the root area of the Prodigy installation on the host computer. Press the **Browse...** button to display a standard Windows browser that will allow you to choose the remote file path required. This can be a mapped drive or a UNC (Universal Naming Convention) path specification. You will most likely browse to a directory named "Prodigy" on the host machine or to the mapped drive that is mapped onto this Prodigy directory.

Using a mapped drive is best as it allows Workstation to detect access problems better and at an earlier stage.

- **Delay.**
This is a timing parameter to prevent overload of the signal manager when the Workstation's network driver starts up. The delay is in milliseconds and is an interval that the system waits between sending all of the current values across at startup time. It tends to be a bit "trial and error" finding the right value, but it is not really needed at all for databases that are not huge.
- **Cache.**
If this value is set then it indicates that new values should only sent by the network driver every n seconds (where n is the value entered). It reduces network traffic in instances where many signals change very rapidly, at the expense that some value states may be missed, as only the latest change is sent. For example, if Cache is set to 2, values are sent every 2 seconds. A given value may start at 1, go to 2, then 3, then 5, then 1 all in that 2-second period. To all intents and purposes it will appear that the signal value didn't change. You should therefore choose whether or not to using caching, and what its value should be, carefully.

- **Unknown.**
Occasionally there may be network problems that result in data not being sent via the network driver. Most often these problems are intermittent and fleeting and have no noticeable effect. Occasionally though they may be longer lasting, and result in no data being transmitted for several seconds. In these instances you may want to indicate the fact that the signal values are no longer necessarily correct and you do this by setting a value in the **Unknown** field. This is the number of seconds that will elapse between the last values being received by the network driver and the system turning all the signal values to *Unknown*.

The Menu Bar

The menu bar that is displayed is separate for each different connection, allowing a different set of facilities to be configured for each different connection. Initially, the first time you ever connect to a given server, a copy is taken of its menu bar. You can the tailor this initial menu bar as required, adding or removing facilities and applying any level of security to items within the menu.

If any other menu bar properties, such as auto-hide or protection against changes are made, these will be common to all menu bars displayed on the Workstation. Only the items displayed on the menu differ from connection to connection.

Prodigy Web Viewer

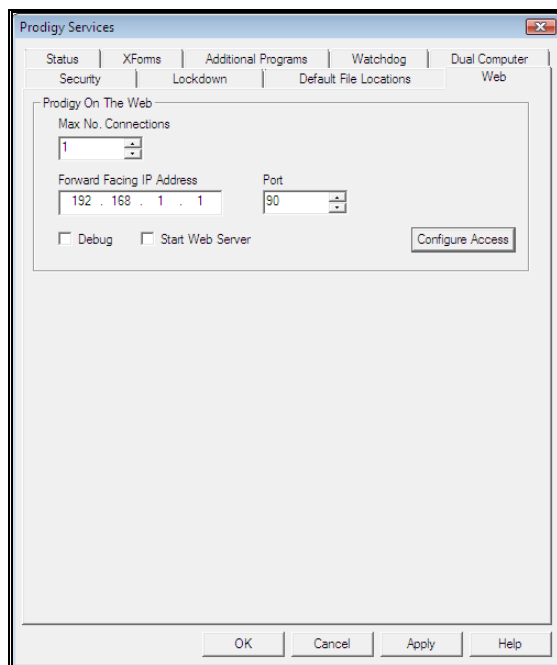
What Is Prodigy Web Viewer

Prodigy Web Viewer allows Prodigy Displays to be viewed across the internet through Microsoft Internet Explorer. It allows massive flexibility in where a Prodigy system can be viewed as a PC that has never had Prodigy installed can be used to observe the state of a system.

Web Viewer Configuration

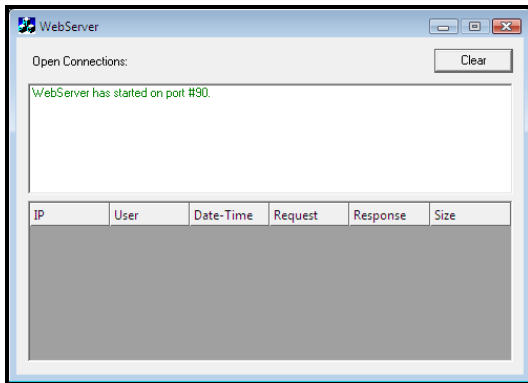
There are two areas within Prodigy that need to be configured in order to set up the web viewer. The first is the Web Tab of the Prodigy Control Panel and the second is the User Configuration. The Features of the Web Tab are detailed below.

Web Tab



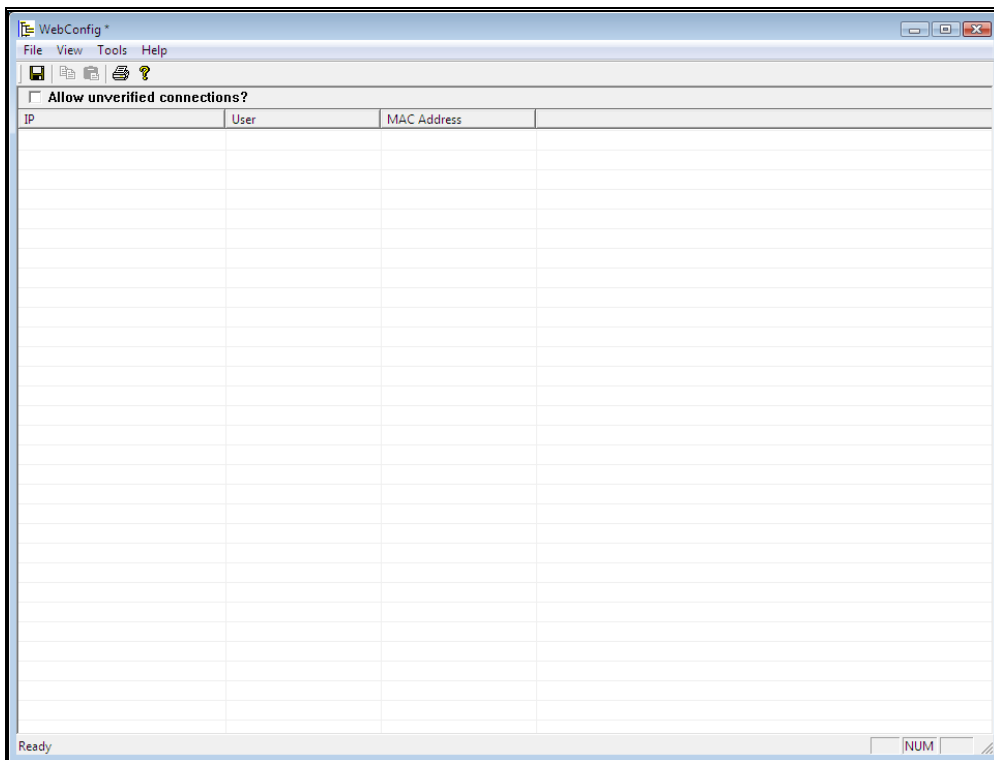
- **Max. No. Connections.**
This allows you to set the maximum number of connections that can be made to Prodigy Web Viewer at any one time. The maximum number is defined in your Prodigy licence and if more people try to connect than the maximum number the last person who attempts will not be allowed a connection until another user disconnects.
- **Forward Facing IP address.**
This is the IP address that an external user will use in order to gain access the Prodigy displays through the internet. If your Prodigy system connects to the internet over a network it will most likely be the IP address the router shows to the outside world.
- **Port**
This is the Port that Prodigy will use to allow external users to access the displays that are available; port forwarding will need to be set up on your router to allow an external computer to see the Prodigy system.
- **Debug and Start Web Server**
The Debug check box provides a debug window that allows the user to observe actions taking place in relation to the Web Server (The web server would still run in the background if this wasn't selected). When Debug is checked, the following window can be seen that also shows which users

are logged on. Start Web Server sets the web server running so that it can be accessed through the IP address and port number.



- **Configure Access**

This option bring up the dialog that can be seen below, it allows you to configure access as to who can use the web server and who cannot. If the **allow unverified connections** box is checked anyone who knows the IP address and port number can access the Prodigy Displays from any computer assuming they know a Username and Password. If access is required from any PC at any time then allow unverified connections should be selected however if access is only required from selected PCs with fixed IP addresses then the appropriate IP addresses should be configured.



User Configuration

Log on name	First Name(s)	Last Name	SMS Number	Act	Email Address	Act
Default User	Default	User		N		N
Plant Manager	Gordon	Smith	+447798345123	Y	gordon.smith@g...	Y

Buttons: Add, Delete, Copy, Security

Log On:
Log on Name: Plant Manager
First Name(s): Gordon
Last name: Smith
Password: []

SMS Alarms:
Telephone Number: +447798345123
SMS Alarm Sets: []
 Active
Restart

Email Alarms:
Address: gordon.smith@goldfromlead.com
Email Alarm Sets: []
 Active
Restart

Web Viewing:
Initial Display: []
Browse...

Privileges:

Privilege	Reason	Allowed
Accept Alarms	No	Yes
Alter Advanced Alarm Properties	No	Yes
Alter Alarm Properties	No	Yes
Configure Access Databases	No	Yes
Configure Alarm Display	No	Yes
Configure Alias Tags	No	Yes
Configure Archive	No	Yes
Configure ATS	No	Yes

Buttons: OK, Cancel, Help

Users must be configured on a Prodigy system in order for Web Viewer to function, this is because a Username and Password are required to log onto the system when it is accessed through Internet Explorer. The Username and Password are the same as those configured in User Configuration; users should also have the Web Viewing privilege so that they can access displays through the internet.

The Web Viewing section needs to have the initial display that the user will see selected; this is done by pressing the Browse button and choosing an appropriate display.

Steps to Set Up and Connect to Prodigy Web Viewer

In order to use Prodigy Web Viewer the following steps should have been taken in order to prepare the system for the task.

- The Web Tab should have been set up on the control panel to tell Prodigy what the forward facing IP address is and what port will be used to access the system.
- Users should have been configured to have the Web Viewing privilege and the Initial Display should have been selected in the User Configuration program.
- If you are connecting to the internet through a router it must be configured to provide port forwarding so that the prodigy system can be seen when the IP address and Port number are typed into internet Explorer.
- Remove any bitmap objects from displays that are to be used because at present Web Viewer cannot transfer bitmap objects. Depending on the complexity of displays in your system it might be useful to create some simpler displays for use with the web viewer and store these in a separate folder in the prodigy configuration area.
- To connect to the Prodigy Web Viewer simply open up an internet explorer window on a PC with an internet connection. Type in the Web address `http://<IP Address>:<Port Number>` where <IP address> and <Port Number> are substituted with the information of your system (the prodigy web server is at `http://82.173.240:90`). You will be presented with a web page with a link to some instructions that detail some modifications you need to make to the internet options, make the modifications and press the link to confirm that they have been made. Now press the Run the Web Viewer button and log onto the Prodigy system that has been set up to run Web Viewer.

Appendices

Drivers

See the separate document **Prodigy User Guide Appendix** on your Prodigy CD.

SLANG Extensions

See the separate document **Prodigy User Guide Appendix** on your Prodigy CD.

System Administration Tasks

See the separate document **Prodigy User Guide Appendix** on your Prodigy CD.

Run Time Parameters

See the separate document **Prodigy User Guide Appendix** on your Prodigy CD.

Communications Wiring

See the separate document **Prodigy User Guide Appendix** on your Prodigy CD.

Index

- 21 CFR Part 11, 199
- Accepting Alarms, 168
- Access Database, 192, 350
- Add A Comment, 261
- Alarm Display, 167
- Alarm History Report. *Superseded: use Alarm History Viewer*
- Alarm History Viewer, 184
- Alarm Schedule, 348, 373
- Alarm Strategy, 45
- Alias Tags, 104, 139, 219, 311, 365
- Animation, 117
- Archiving, 502
- ATS, 193, 368
- Audit Trail, 199, 205, 212
- Audit Trail Viewer, 212
- Automated Test Sequencer, 193, 368
- Backup, 502
- Batch Chooser, 258
- Batch Data Recorder, 354
- Batch Handling
 - Access Database Configuration, 350
 - BDR, 354
 - Down Time Reports, 400
 - Easy Report, 271
 - Historical Trending, 107, 143
 - Proof Of Process Report, 305
 - XForm, 225
- BDR, 354
- Black Box Recorder, 51, 490
- Calibration, 49, 481
- CFR, 199
- Chartstation. *See Workstation*
- Comments, 46, 146, 261
- Configuration
 - Access Database Configuration, 350
 - Alias Tags, 219
 - Automated Test Sequencer (ATS), 368
 - Automatic Archiving, 502
 - Batch Chooser Configuration, 258
 - Batch Data Recorder (BDR), 354
 - Data Processing Report Configuration, 404
 - Data Processor Configuration, 399
 - Display Builder, 68
 - Display Launcher, 187
 - Down Time Logger Configuration, 375
 - Down Time Reason Configuration, 383
 - Driver Configuration, 19
 - Easy Report, 271
 - Email Manager, 182
 - Multigraph, 316
 - Multigraph Selection, 326
 - OPC Server Configuration, 459
 - Proof Of Process Report, 305
 - Shop Floor Terminal Configuration, 406
 - Signal Database Configuration, 24
 - SLANG Editor, 426
 - SMS Manager Configuration, 179
 - Software Controllers Configuration, 470
 - Timed Report Configuration, 478
 - Users Configuration, 205
 - Web Upload Configuration, 510
 - XForm Edit, 227
- Control Panel, 191
- Control Sequence. *See SLANG*
- Controllers. *See Software Controllers*
- CSV, 151, 263, 331, 339, 489
- Data Archiving, 502
- Data Backup, 502
- Data Processing Report Configuration, 404
- Data Processor Configuration, 399
- DDE, 466
- Default File Locations, 201
- Disable, 185, 192, 222, 414, 487
- Display Builder, 68, 138, 188, 221, 223, 269
- Display Launcher, 187
- Down Time Logger, 381
- Down Time Monitoring, 375
 - Data Processing Report Configuration, 404
 - Data Processor Configuration, 399
 - Down Time Logger, 381
 - Down Time Logger Configuration, 375
 - Down Time Overview, 395
 - Down Time Reason Configuration, 383
 - Down Time Viewer, 388
 - Shop Floor Terminal Configuration, 406
- Down Time Overview, 395
- Down Time Viewer, 388
- Driver, 19, 28, 38
- Driver Configuration, 19, 490
- DTM02 Terminals. *See Shop Floor Terminals*
- DTM03 Terminals. *See Shop Floor Terminals*
- Dual Computer, 197, 512
- Easy Report, 271
- Email, 272, 304, 306
- Email Alarms, 176
- Email Reports, 180
- Event Logger, 193, 487
- Event Text, 41
- Event Viewer, 487
- Expert Data System, 494
- Export, 63, 151, 270, 394, 488
- Forms. *See XForm*
- Heatmap, 71
- Historical Trending, 106, 138, 140, 270
- HTML Reports, 509
- Import, 32, 63, 352
- Inverting Signals, 51
- License Installer. *See Getting Started With Prodigy guide, available on the CD*
- License Registration. *See Getting Started With Prodigy guide, available on the CD*
- Lockdown, 201
- Log Off, 209
- Log On, 209
- Machine Utilisation. *See Down Time Monitoring*

- Menu Bar, 155
- Multigraph, 314
- On-line Maintenance, 93
- On-line Media, 93
- OPC, 452
- OPC Client (standalone), 461
- PDF, 201, 272, 304, 306, 357
- PID Controllers. *See Software Controllers*
- Profile Curves, 263
- Profile Editor, 263
- Programming Language. *See SLANG*
- Proof Of Process Report, 221, 304, 305
- Quick Start, 204
- Range, 41
- Real Time Trending, 134
- Recipe Handling. *See Batch Handling*
- Recording Strategy, 44
- Remote Access
 - Email Alarms, 176
 - SMS Alarms, 176
 - Web Viewer, 519
 - Workstation/Viewstation/Chartstation, 514
- Replay, 223
- Report, 87, 180, 257, 271, 305, 314, 357, 394, 399, 404, 478, 486, 509
 - Down Time, 404
 - Easy Report, 271
 - Email, 180
 - Multigraph, 314
 - Proof Of Process Report, 305
 - SMS, 180
 - Timed Report, 478
- Resetting Strategy, 43
- SCADA Control Panel, 191
- Screen Saver, 210
- Security, 199, 205, 515
- Sequence Language. *See SLANG*
- Shop Floor Terminals, 385, 405
- Signal Calibration. *See Calibration*
- Signal Database, 24, 37, 53, 59, 63
- Signal Database Export, 63
- Signal Database Import, 63
- Signal Disabler, 222
- Signal Monitor, 24, 65
- Signal Selection, 215
- Signal Smoothing, 50
- SigScan, 215
- SLANG, 168, 197, 374, 411
- SLANG Control Panel, 445
- SLANG Extension, 438, 492, 523
- SMS Active Alarms, 178
- SMS Alarms, 176, 178
- SMS Reports, 180
- Software Controllers, 193, 468
- SPC, 153, 327, 347
- SQL Server, 192
- Starting Additional Programs, 195
- Starting Prodigy, 187, 191
- Statistical Process Control. *See SPC*
- Stopping Prodigy, 203
- Thermal Image, 71
- Timed Reports, 478, 509
- Trend, 41, 68, 71, 104, 124, 134, 138, 147, 221, 258, 261, 263, 305, 327
- Trending
 - Exporting data to CSV/Excel, 151
 - Historical, 138
 - Real Time, 134
- Units, 41
- Update Rate, 40
- Users Configuration, 205
- Viewstation. *See Workstation*
- Watchdog, 196, 202
- Web Upload, 188, 509
- Web Viewer, 202, 519
- Workstation, 189, 514
- Workstation Connect, 514
- XForm, 189, 194, 225, 354, 367