# *Whisker: in pictures*

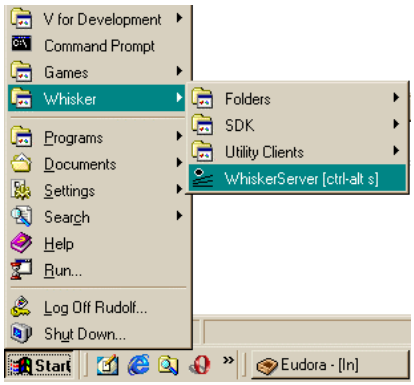
*www.whiskercontrol.com*

15 Feb 2006

## Overview


*Whisker_logo_for_manual.tif*

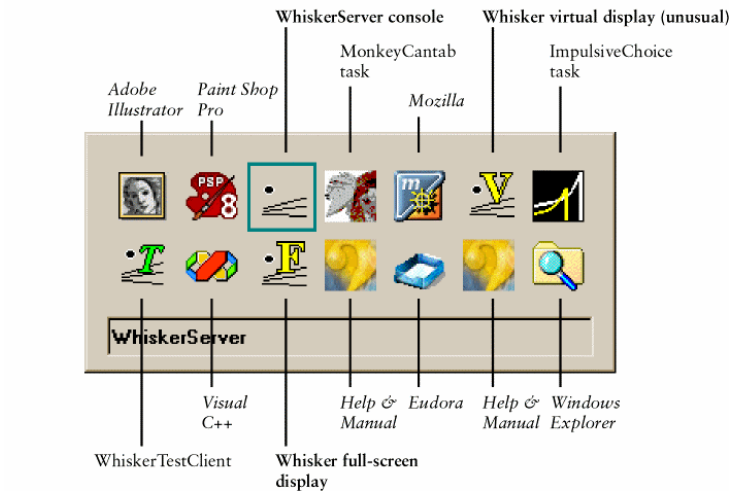
**Whisker is installed under Windows.**




*Start_Menu_With_Whisker.bmp*

**You run the Whisker server (which talks to the hardware) and Whisker clients (which implement behavioural tasks), together with other programs in Windows.**

*Example of pressing Alt-Tab to switch windows while running Whisker*



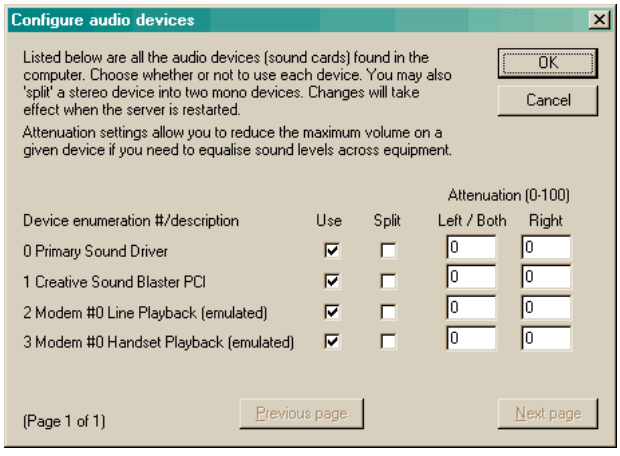

*AltTab_Key.BMP*

## Hardware

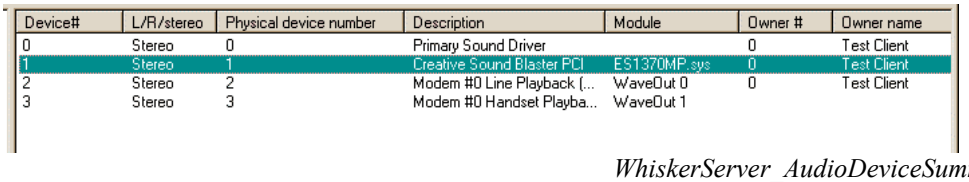
**Whisker supports a wide range of hardware, including digital I/O devices from a range of manufacturers…**

**… touchscreens via the Touch-Base UPDD drivers and any serial port supported by Windows…**

**… any sound card supported by Windows (also allowing the splitting of a single stereo device into two mono sound devices)…**



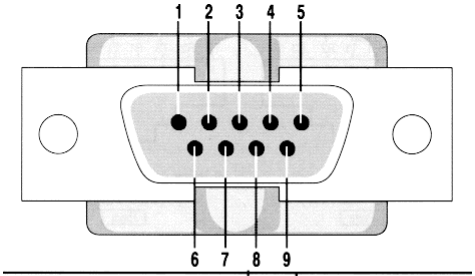

*WhiskerServer_ConfigureAudioDevices.bmp*



| Device# | L/R/stereo | Physical device number | Description | Module | Owner # | Owner name |
|---|---|---|---|---|---|---|
| 0 | Stereo | 0 | Primary Sound Driver | | 0 | Test Client |
| 1 | Stereo | 1 | Creative Sound Blaster PCI | ES1370MP.sys | 0 | Test Client |
| 2 | Stereo | 2 | Modem #0 Line Playback (... | WaveOut 0 | 0 | Test Client |
| 3 | Stereo | 3 | Modem #0 Handset Playba... | WaveOut 1 | | |

*WhiskerServer_AudioDeviceSummaryView.bmp*

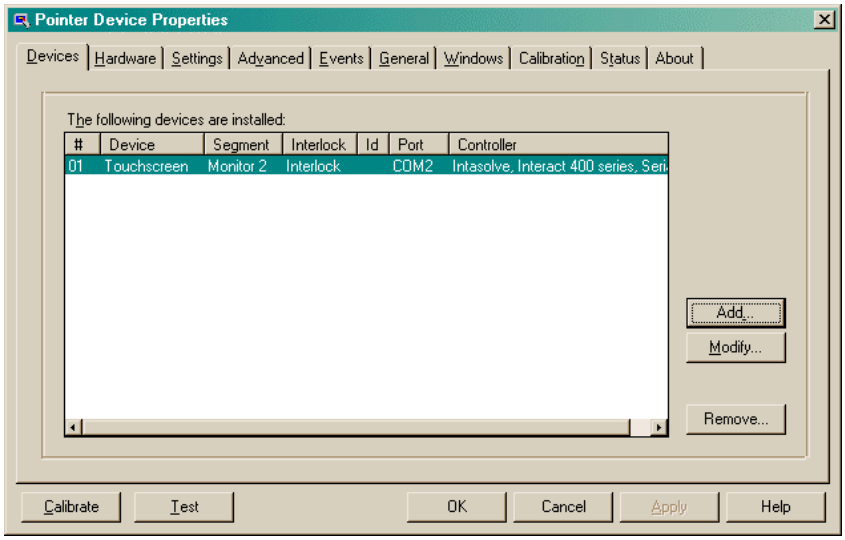
**… any video card supported by Windows, including multimonitor cards…**
**… and serial ports used as digital I/O devices.**




| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | Data Carrier Detect | 6 | Data Set Ready |
| 2 | Received Data | 7 | Request to Send |
| 3 | Transmitted Data | 8 | Clear to Send |
| 4 | Data Terminal Ready | 9 | Ring Indicator |
| 5 | Signal Ground | | |

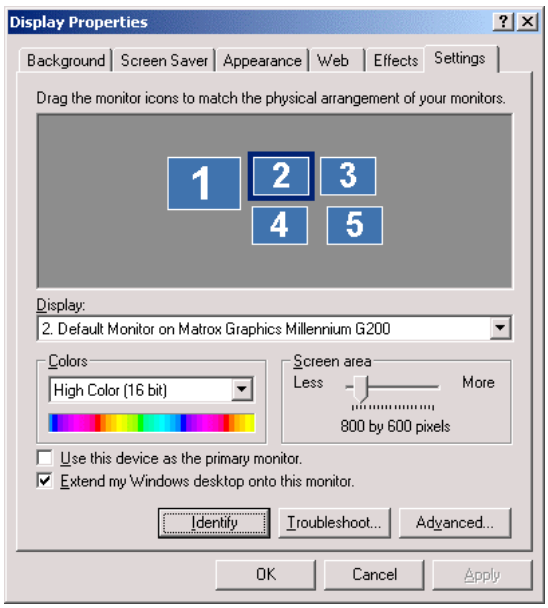
*Serial_9Pinlabels.BMP*

**Whisker uses the Touch-Base UPDD driver to communicate with touchscreens from any manufacturer supported by UPDD.**

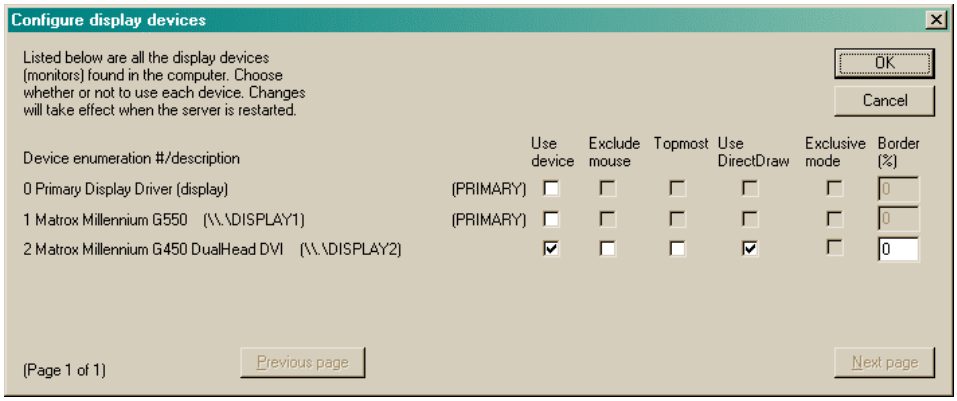



*UPDD_v3_Devices.bmp*

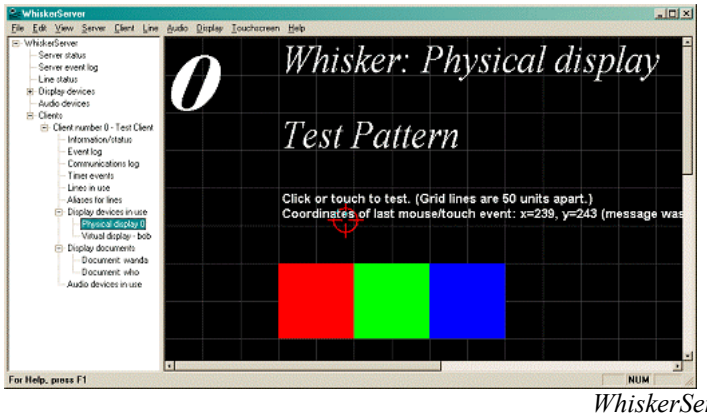
**Whisker uses the Windows multimonitor facility to drive multiple monitors (with touchscreen attached, if desired).**




*Windows_ConfigureMultimonitor.bmp*

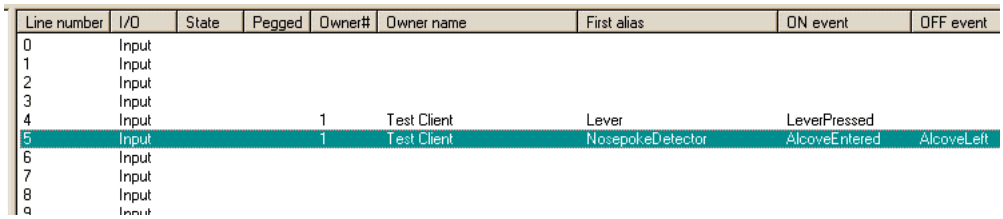



*WhiskerServer_ConfigureDisplays.bmp*




*WhiskerServer_ConsoleViewOfLargeDisplay.bmp*

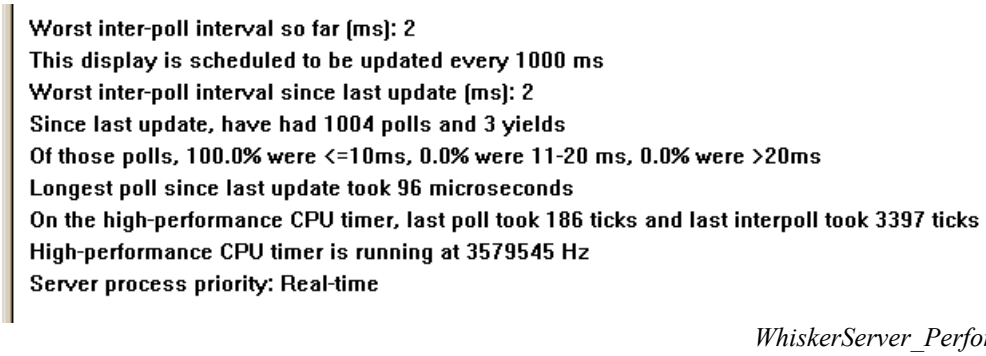
**Whisker allows you to manipulate the on/off state of digital input/output lines directly to test your equipment, and to watch the state of all lines (or all lines being used by a particular task) as tasks are running.**



| Line number | I/O | State | Pegged | Owner# | Owner name | First alias | ON event | OFF event |
|---|---|---|---|---|---|---|---|---|
| 0 | Input | | | | | | | |
| 1 | Input | | | | | | | |
| 2 | Input | | | | | | | |
| 3 | Input | | | | | | | |
| 4 | Input | | | 1 | Test Client | Lever | LeverPressed | |
| 5 | Input | | | 1 | Test Client | NosepokeDetector | AlcoveEntered | AlcoveLeft |
| 6 | Input | | | | | | | |
| 7 | Input | | | | | | | |
| 8 | Input | | | | | | | |

*WhiskerServer_LineStatusView_SelectingALine.bmp*

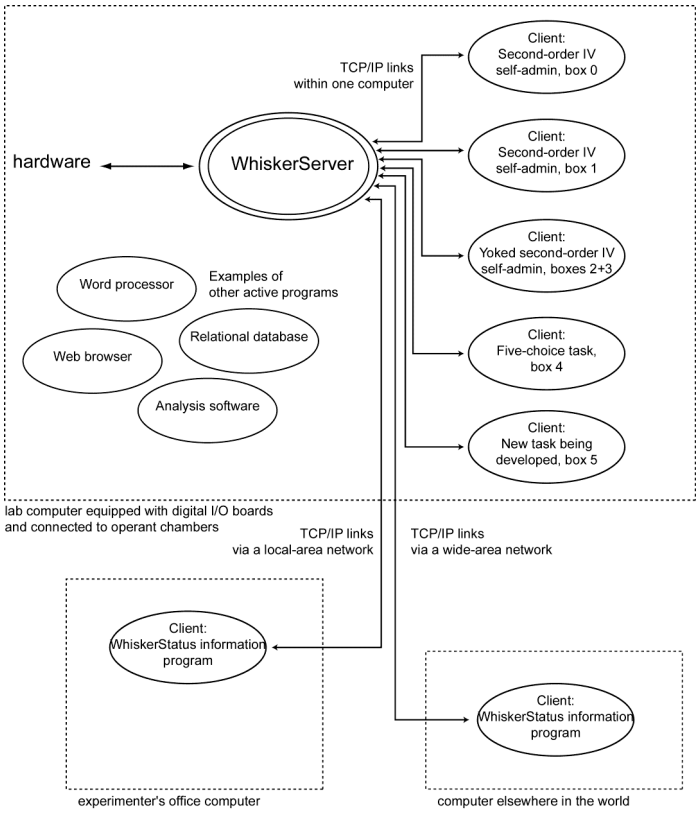
**Whisker monitors its own performance, taking care to aim for 1 kHz hardware polling (1 ms temporal resolution) while never consuming more than 50% of CPU time.**




*WhiskerServer_Performance.bmp*

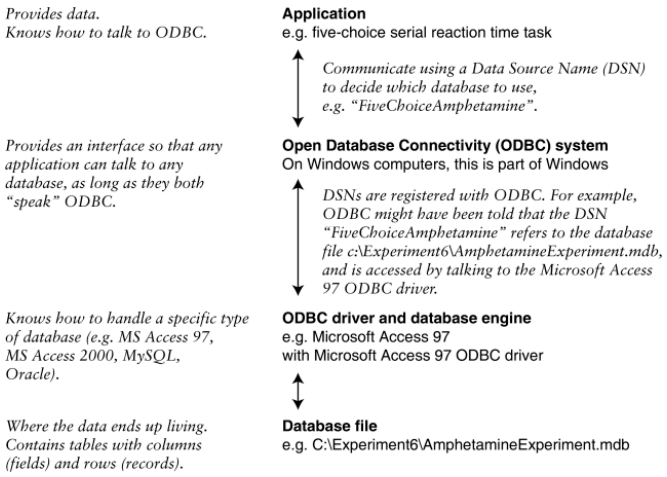
# Clients

**Whisker uses a client–server architecture.**



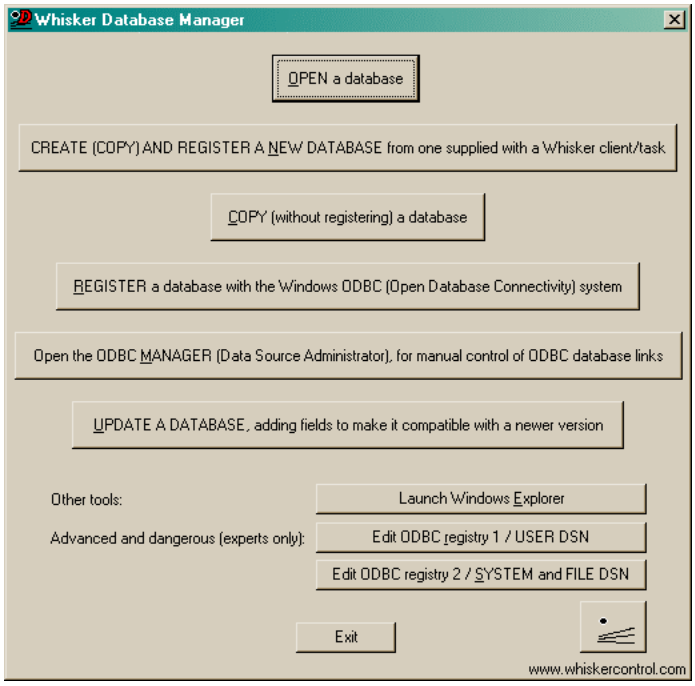

*Whisker_software_schematic.bmp*

**Whisker tasks typically write their data direct to an ODBC-compatible database.**



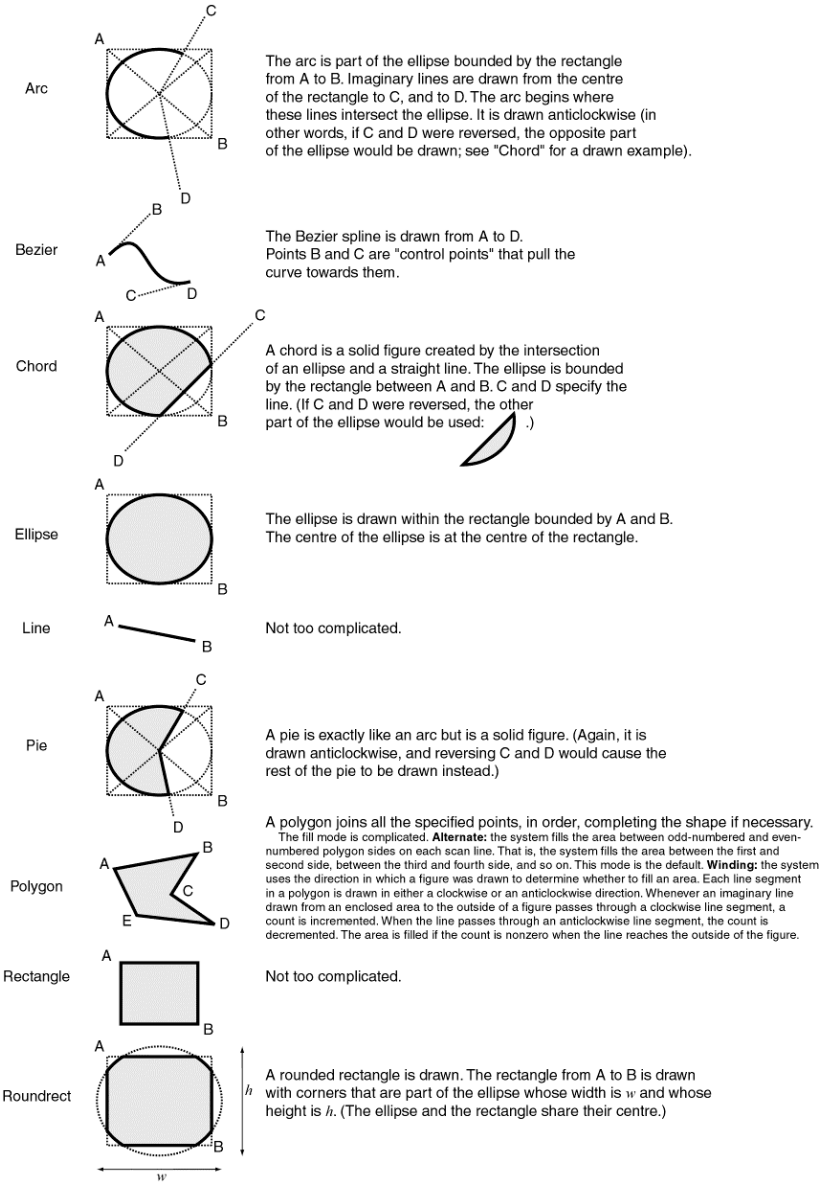

*Principles_of_ODBC.BMP*

**The Whisker Database Manager simplifies the use of ODBC-compatible databases (such as Microsoft Access).**



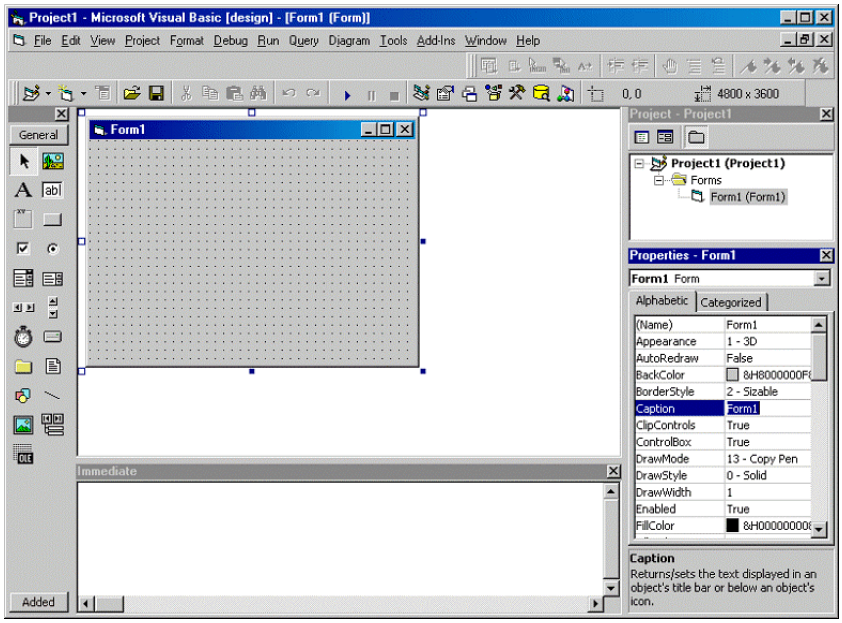

*DatabaseManager_Main.bmp*

**Stimuli can be drawn using any of the Windows drawing primitives, through a simple interface.**

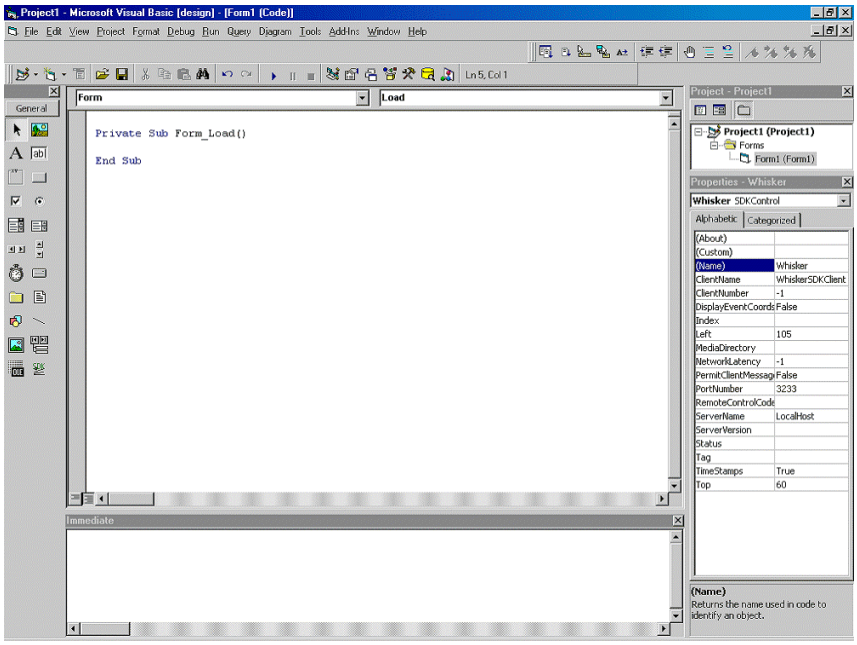



*Windows_GDI_drawing.BMP*

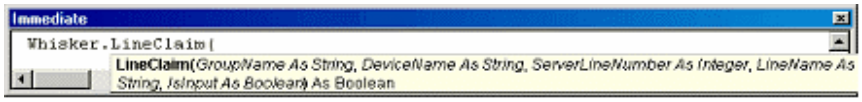
**The Whisker Visual Basic Software Development Kit makes the process of developing Whisker tasks easy.**




*Tutorial1_NewProject.bmp*




*Tutorial1_FormLoad.bmp*



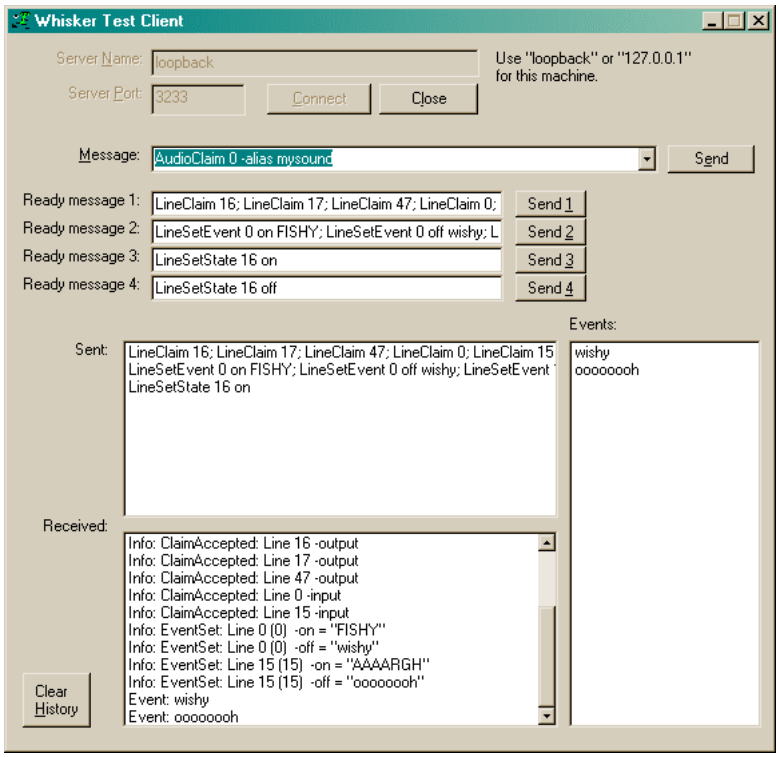

*Tutorial2_Licker_UI_2.bmp*


*LineClaim_SDKMethod.bmp*


*LineReadState_SDKMethod.bmp*


*LineSetState_SDKMethod.bmp*

**Whisker uses a high-speed text-based network communication system. A test client allows you to explore what happens behind the scenes.**




*WhiskerTestClient.bmp*

**Clients can assign names to individual input/output (I/O) lines.**

| Line number | Alias |
|---|---|
| 48 | HOUSELIGHT |
| 49 | LEFTLIGHT |
| 50 | RIGHTLIGHT |
| 50 | STIMULUS |
| 51 | TRAYLIGHT |
| 52 | PUMP |
| 52 | REINFORCER |
| 53 | DIPPER |
| 54 | LEFTLEVERCONTROL |
| 54 | INACTIVELEVERCONTROL |
| 55 | RIGHTLEVERCONTROL |
| 55 | ACTIVELEVERCONTROL |
| 72 | NOSEPOKE |
| 73 | LEFTLEVER |
| 73 | INACTIVELEVER |
| 74 | RIGHTLEVER |
| 74 | ACTIVELEVER |
| 75 | LOCOBEAM_FRONT |
| 76 | LOCOBEAM_MIDDLE |
| 77 | LOCOBEAM_REAR |

*WhiskerServer_ClientAliasesForLines.bmp*

**Communication between the client and the server can be monitored for debugging a new client.**

| Time | Source | Message |
|---|---|---|
| 18:17:26 | Server | Event: Locomotor_Middle |
| 18:17:26 | Server | Event: Locomotor_Front |
| 18:17:26 | Server | Event: Locomotor_Rear |
| 18:17:27 | Server | Event: Locomotor_Front |
| 18:17:32 | Server | Event: Active_Lever |
| 18:17:32 | Client | ReportStatus Box 5 (m11) - active 482, inactive 58, stim 46, reinf 4 - Task started. |
| 18:17:32 | Server | Event: Active_Lever |
| 18:17:32 | Client | ReportStatus Box 5 (m11) - active 483, inactive 58, stim 46, reinf 4 - Task started. |
| 18:17:32 | Server | Event: Active_Lever |
| 18:17:32 | Client | SetState STIMULUS on |
| 18:17:32 | Client | SetState HOUSELIGHT off |
| 18:17:32 | Client | RequestTimerEvent 1000 0 _sys51 |
| 18:17:32 | Server | TimerCreated: duration 1000, reloadcount 0 |
| 18:17:32 | Client | RequestTimerEvent 1000 0 Unit_Schedule_Finished |
| 18:17:32 | Server | TimerCreated: duration 1000, reloadcount 0 |
| 18:17:32 | Client | ReportStatus Box 5 (m11) - active 484, inactive 58, stim 47, reinf 4 - Task started. |
| 18:17:33 | Server | Event: Unit_Schedule_Finished |
| 18:17:33 | Server | Event: _sys51 |
| 18:17:34 | Server | Event: Locomotor_Middle |
| 18:17:34 | Server | Event: Locomotor_Rear |

*WhiskerServer_ClientCommsLog.bmp*

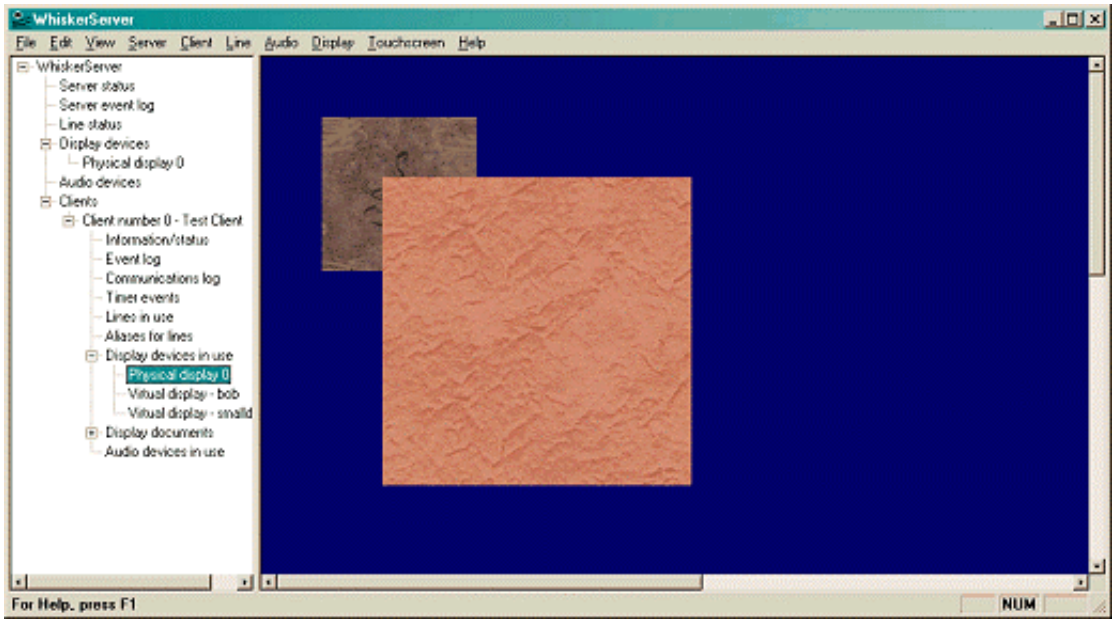
**For visual displays, clients can manage several documents, each showing different content, and choose which**

| Document# | Document name | Number of objects |
|---|---|---|
| 5 | CS_plus | 1 |
| 6 | CS_minus | 1 |
| 7 | CS_both | 2 |

*WhiskerServer_ClientDocumentSummary.bmp*

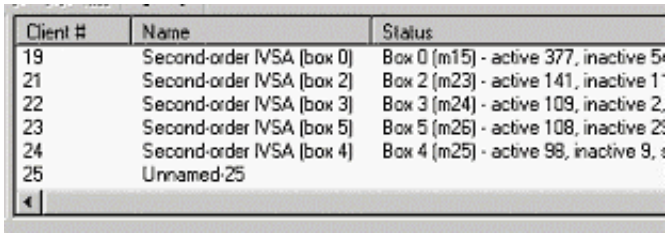
**document to show to the subject.**

**The server can be used to monitor what any subject is seeing.**



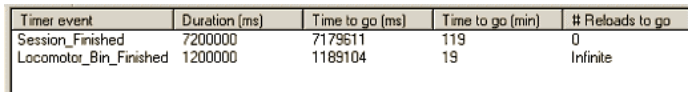

*WhiskerServer_ClientIndividualDisplay.bmp*

**The server allows many clients (tasks) to run simultaneously, and allows you to keep an eye on all of them.**



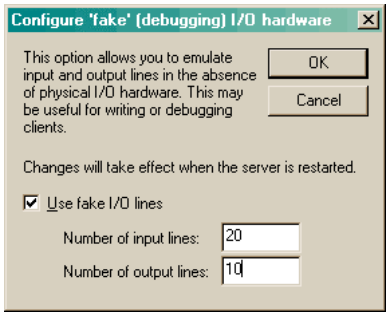

*WhiskerServer_ClientSummaryView.bmp*

**Clients can display things on a screen, respond to touchscreen, mouse, and keyboard events, play sounds, turn devices on and off, and create timers.**




*WhiskerServer_ClientTimerEvents.bmp*

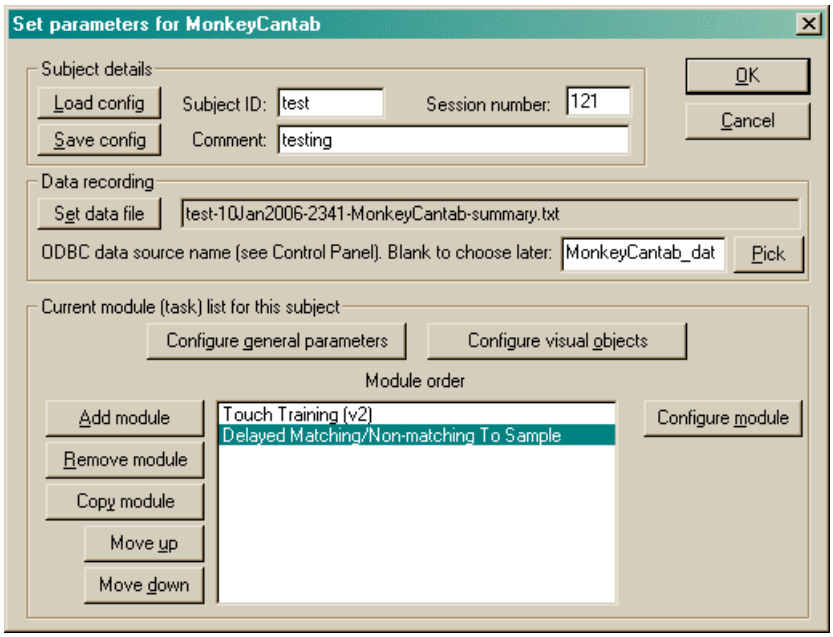
**Whisker provides advanced debugging features, including the facility of 'fake' devices (digital I/O, audio devices, etc.), allowing you to write tasks and test them on computers that do not have digital I/O installed.**



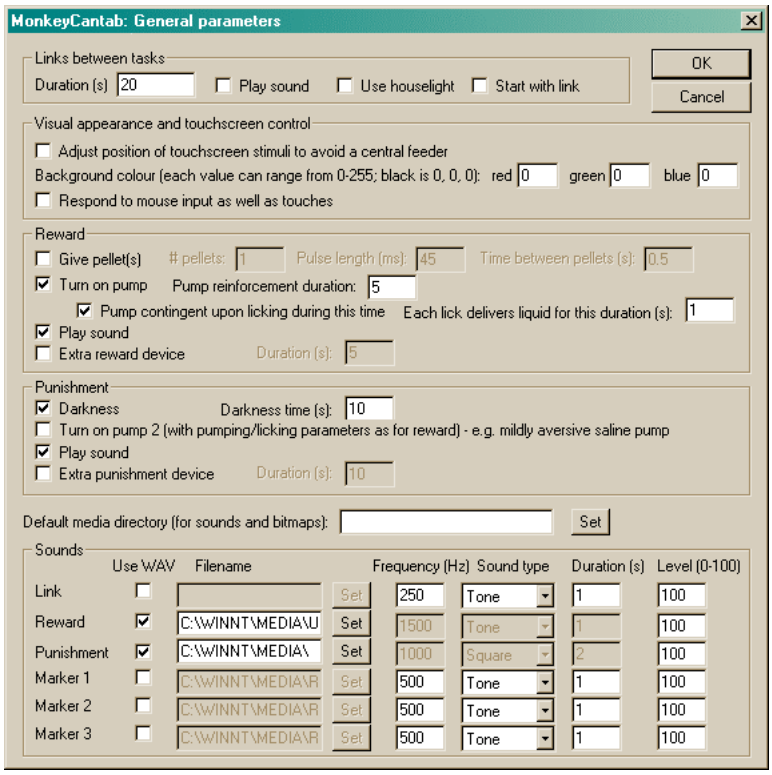

*WhiskerServer_ConfigureFakeLines.bmp*

# Existing tasks

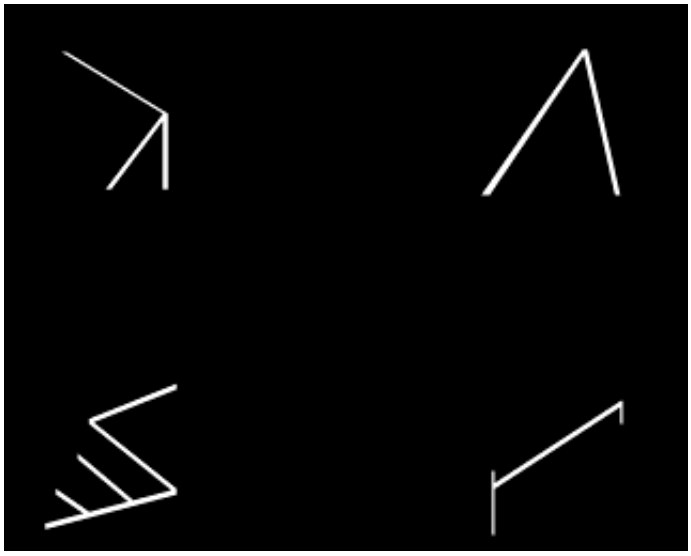
**Tasks written for Whisker to date include some of the most popular in their class worldwide, including MonkeyCantab…**



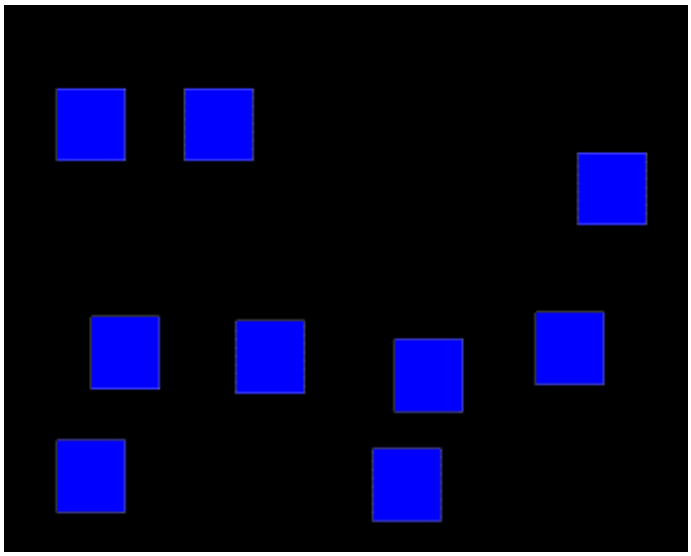

*MonkeyCantab_Parameters.bmp*



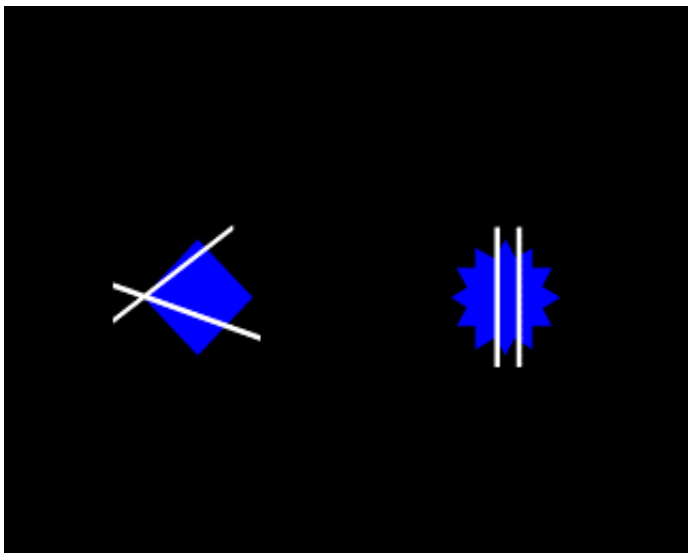

*MonkeyCantab_GeneralParams.bmp*



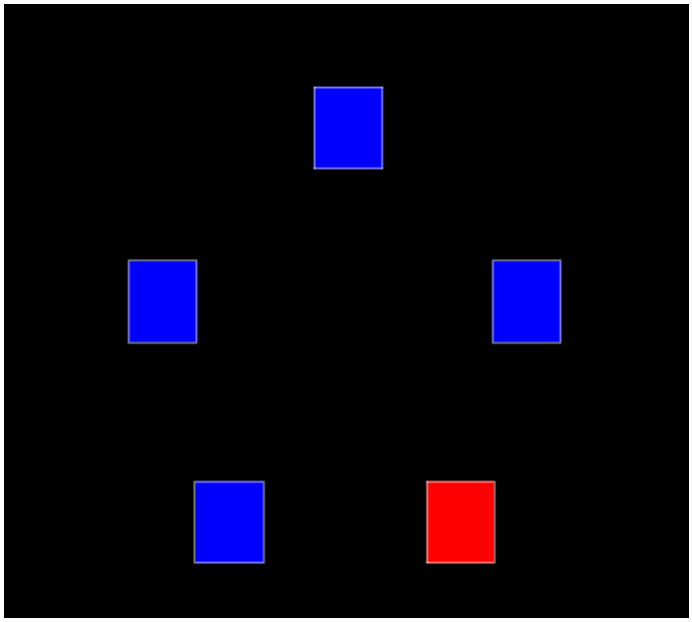
*MonkeyCantab_DMTS_Screenshot2.bmp*



*MonkeyCantab_SWMrunning.bmp*

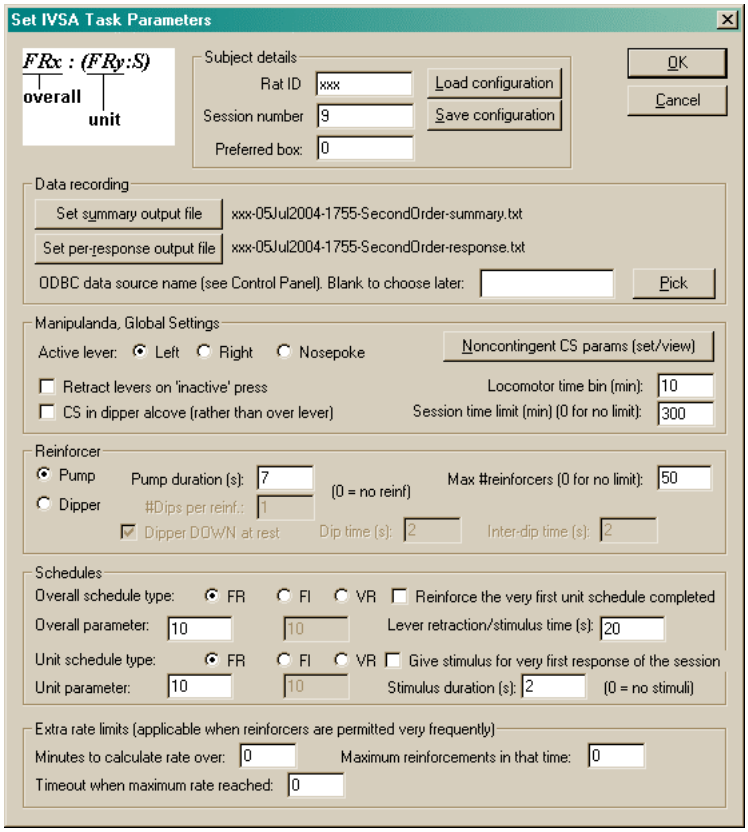


*MonkeyCantab_VDS_Screenshot.bmp*

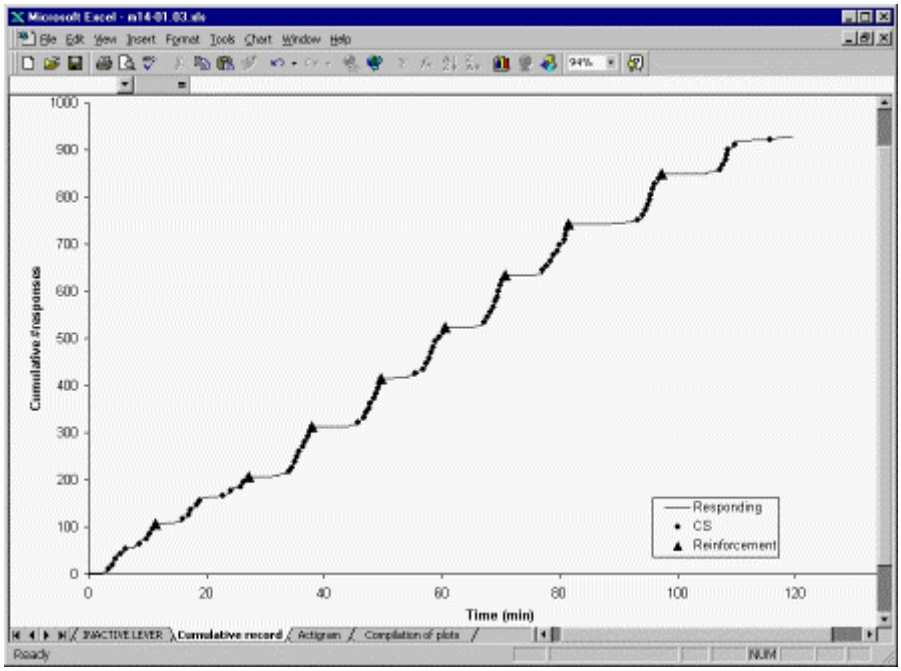


*MonkeyCantab_5C_5wayscreenshot.bmp*

**… second-order schedules of reinforcement…**

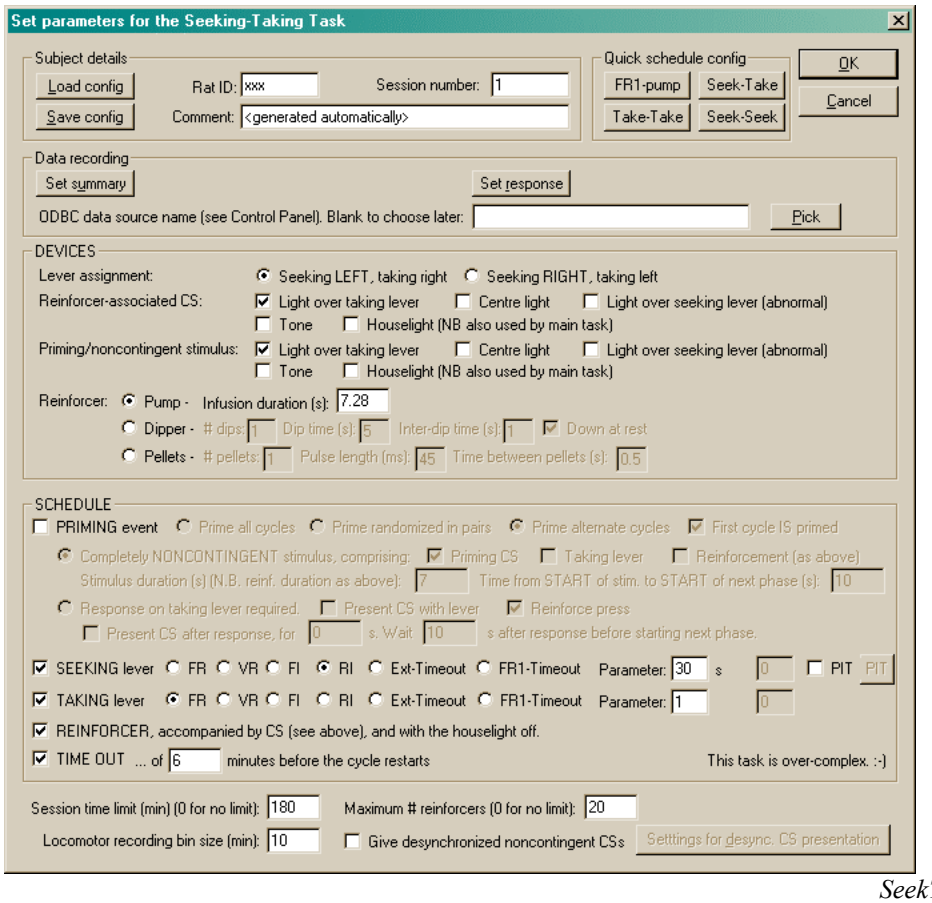



*SecondOrder_Params.bmp*



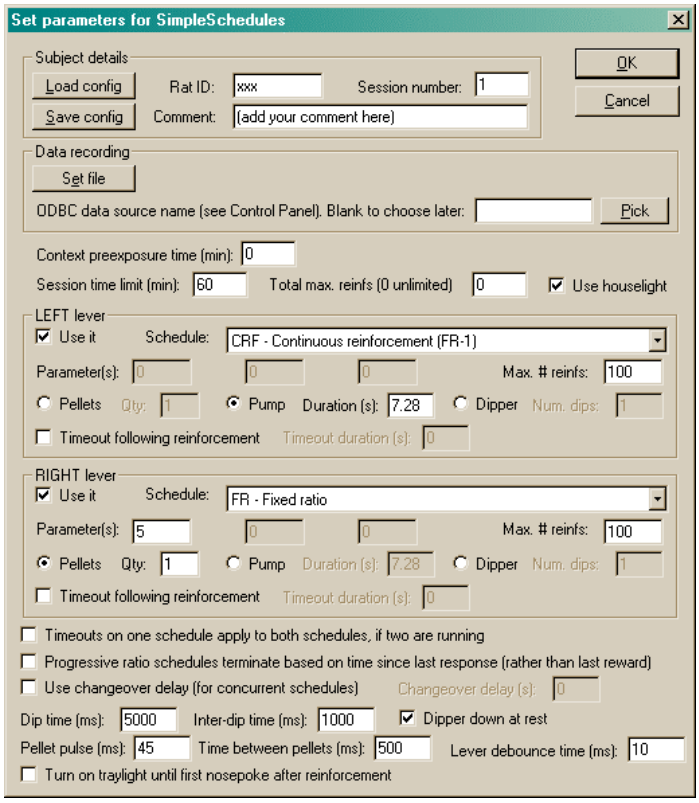

*SecondOrder_NiceExample.bmp*

**… the Seeking–Taking Task…**



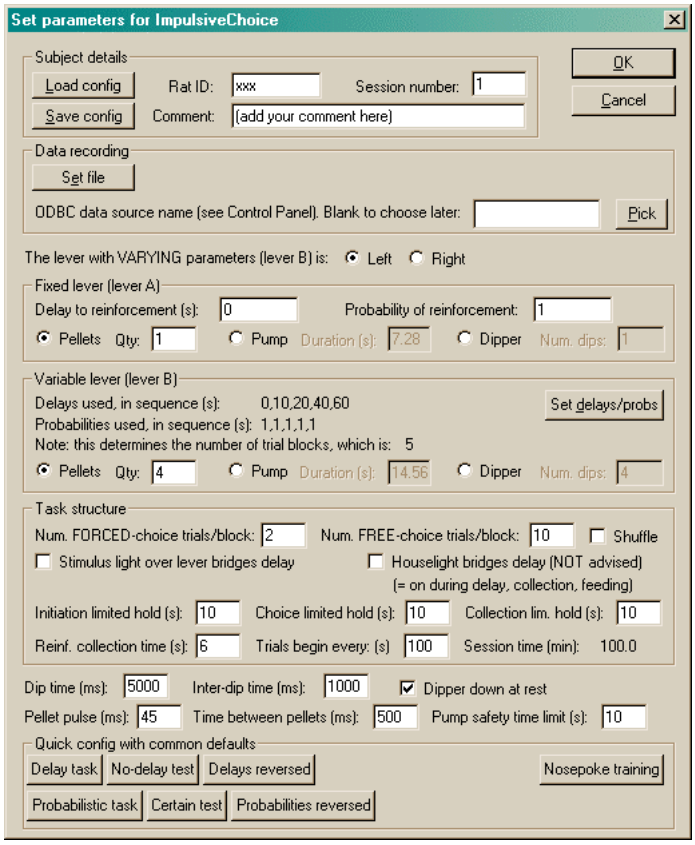

*SeekTake_Params.bmp*

**… simple schedules of reinforcement…**



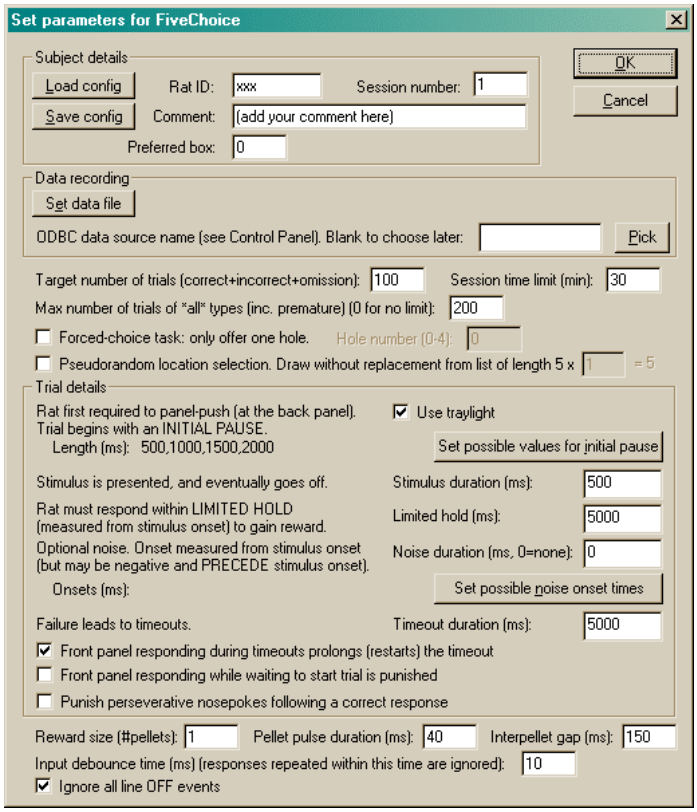

*SimpleSchedules_Parameters.bmp*

**… delayed reinforcement choice tasks…**




*ImpulsiveChoice_Params.bmp*

**… the Five-Choice Serial Reaction Time Task…**




*FiveChoice_Parameters.bmp*

**… and many more.**