

# PIT

## A Whisker client

---

*by Rudolf Cardinal*

*[www.whiskercontrol.com](http://www.whiskercontrol.com)*

*Copyright (C) Cambridge University Technical Services Ltd.*

*Distributed by Campden Instruments Ltd ([www.campden-inst.com](http://www.campden-inst.com))*



# PIT

© Cambridge University Technical Services Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2017 in Cambridge, UK

## **Creator (Whisker)**

*Rudolf N. Cardinal*

## **Design and Programming (Whisker)**

*Rudolf N. Cardinal*

*Michael R. F. Aitken*

## **Legal Advisor (CUTS)**

*Adjoa D. Tamakloe*

## **Sales (Campden)**

*Julie Gill*

## **Contacting the authors:**

*For information about Whisker, visit <http://www.whiskercontrol.com/>.*

*If you have sales enquiries about Whisker, contact Campden Instruments Ltd at <http://www.campden-inst.com/>.*

*If you have comments or technical enquiries that cannot be answered by the sales team, contact the authors:*

*Rudolf Cardinal ([rudolf@pobox.com](mailto:rudolf@pobox.com))*

*Mike Aitken ([m.aitken@psychol.cam.ac.uk](mailto:m.aitken@psychol.cam.ac.uk))*

# Table of Contents

Foreword	1
<b>Part I PIT</b>	<b>2</b>
1 About PIT .....	2
2 Required devices .....	2
3 Using the task .....	4
4 Parameters .....	6
5 Creating a new ODBC source .....	7
<b>Index</b>	<b>12</b>

---

# Foreword

## WARNING

**Whisker is a system designed for research purposes only, and should never be used to control medical apparatus or other devices that could endanger human life.**

## DISCLAIMER

**The authors, copyright holders, and distributors disclaim all responsibility for any adverse effects that may occur as a result of a user disregarding the above warning.**

# 1 PIT

## 1.1 About PIT

### Purpose

Simple Pavlovian-to-instrumental transfer (PIT).

### Software requirements

Requires Whisker v2.0 or greater.

### Data storage

- Text-based output to disk.
- ODBC data storage to a database (supplied).

### Author

Rudolf Cardinal ([rudolf@pobox.com](mailto:rudolf@pobox.com)).

### Copyright

Copyright © Cambridge University Technical Services Ltd

### Version history

- Version 1.1: majority of development done under this number.
- Version 2.0 (Aug-Sep 2002): default ODBC database facility; integrated HTML help
- Version 2.1: fixed minor bug (unsaved dialogue parameters when picking ODBC database)
- Version 2.2 (10 Dec 2002):
  - option to use stimulus as discriminative stimulus rather than true Pavlovian CS in the Pavlovian phase (specifically, as an SD indicating that nose-poking will be reinforced; instead of running an RT schedule, it's an RI schedule of reinforcing nose-poking) as requested by Pat di Ciano.
  - Also, option to have houselight off during CS but on at other times...
  - ... and conversion to XML parameters.
- Version 2.3 (22 Nov 2003): writes version number and compilation date to summary file.
- Version 2.4 (16-21 June 2006) and v2.5 (22 June 2006): more flexible options for the stimuli; more options for neutral stimulus presentation in the Pavlovian phase; more options for the order of the transfer test; option for FI instrumental training.
- Version 2.6 (27 June 2006). Houselight state at the end of the task was "leave as during the task"; now an option in its own right.
- Version 2.7 (8 March 2007). Easier compilation for users.
- Version 3.0 (12 Jan 2009). Server default changed from "loopback" to "localhost" (Windows Vista compatibility and more general standardization).

## 1.2 Required devices

The program requires to claim devices in groups named **box0**, **box1**, **box2...** with device names as listed below in bold:

```
// Names of lines the program expects to be able to claim
NOSEPOKE           // input
LEFTLEVER         // input
RIGHTLEVER        // input
```

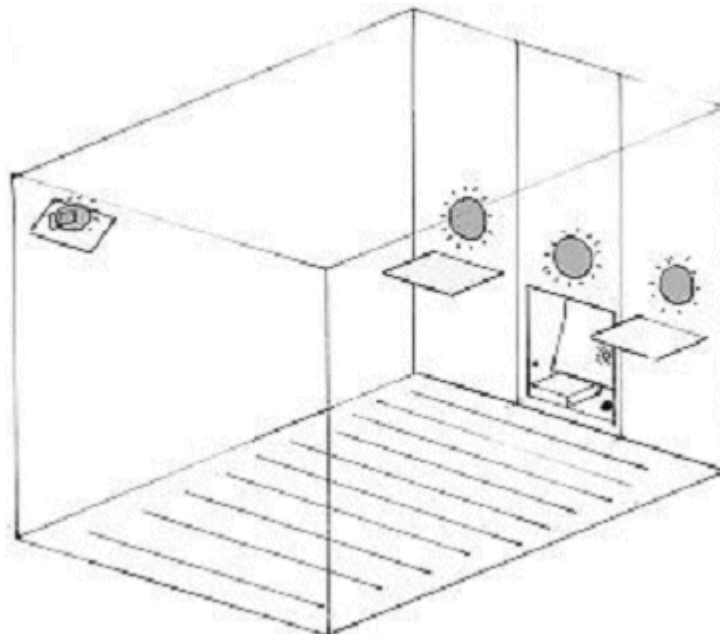
```
LOCOBEAM_FRONT // input
LOCOBEAM_MIDDLE // input
LOCOBEAM_REAR // input
HOUSELIGHT // output
PUMP // output
DIPPER // output
LEFTLEVERCONTROL // output
RIGHTLEVERCONTROL // output
LEFTLIGHT // output
RIGHTLIGHT // output
PELLET // output
TRAYLIGHT // output
CENTRELIGHT // output
CLICKER // output
TONE // output
```

// Aliases used while the program is in full flight, which it therefore expects **not** to be present on the server:

```
StimulusOne
StimulusTwo
ActiveLever
ActiveLeverControl
LeverLight
OppositeLight
```

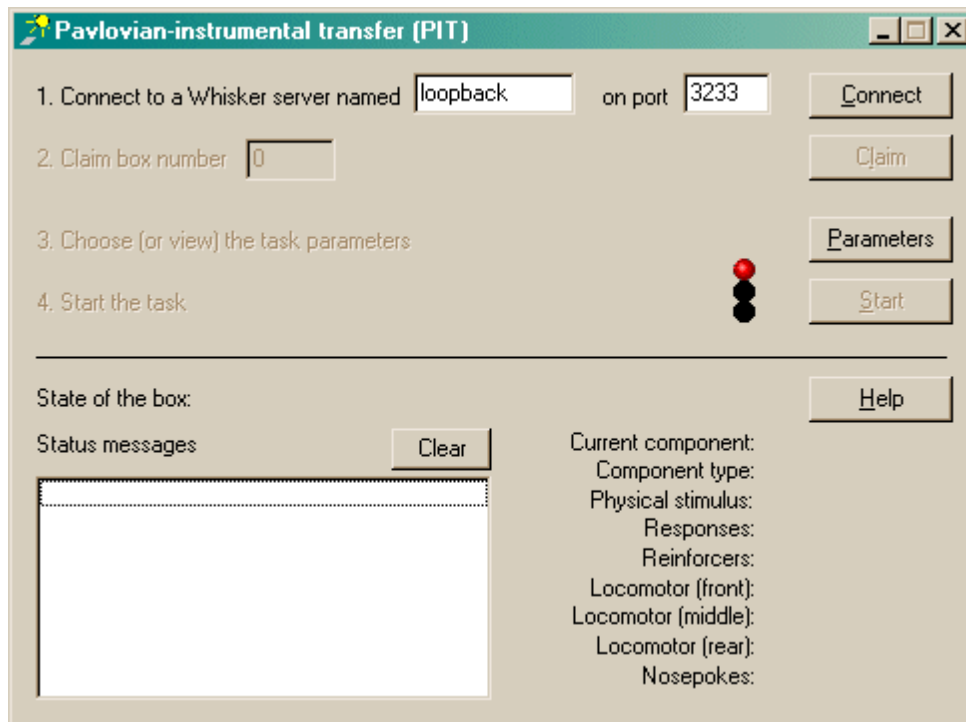
Please ensure that these devices are available and listed in the device definition file in use by the server.

This is the type of operant chamber we're talking about:



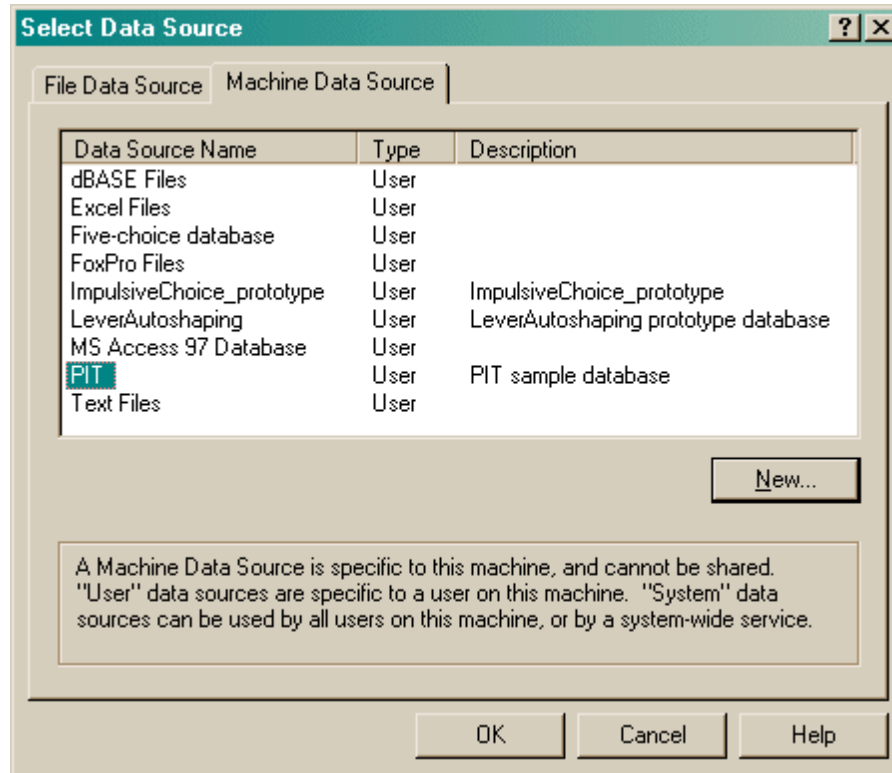
## 1.3 Using the task

When you run the task, the main screen looks as follows:



You must connect to a Whisker server, claim an operant chamber (box), and set up the [parameters](#) for your task. Once that's done, the traffic lights will turn amber. When you are ready, press *Start* to begin the task.

When the task finishes, it saves data to disk and pops up a new dialogue box for you to select a database to store the data to. (The data sources are configured under *Control Panel* → *ODBC*.) If you previously specified an ODBC data source in the parameters, that data source is used automatically and you will only see a dialogue box if something goes wrong and the program needs your input.





## 1.4 Parameters

The parameters dialogue box looks like this:

**Set parameters for Pavlovian-instrumental transfer**

**Subject details**

Load config Rat ID: xxx Session number: 3 ODBC data source name (Leave blank to choose database later.) OK  
 Save config Comment: test Pick Cancel

**More data recording**

Set summary xxx-27Jun2006-1443-PIT-summary.txt Set response xxx-27Jun2006-1443-PIT-response.txt

**Global settings**

Only a single lever is used. Choose  left or  right  Houselight on when no stimuli being presented  
 STIMULI:  Stimulus 1 CS, stimulus 2 NEUTRAL  1 NEUTRAL, 2 CS  Houselight on during CS (ignored if HL is part of CS!)  
 Leave houselight on when task ends  
 Stimulus 1:  Houselight  Centre light  Light over lever  Light on opposite side  Tone  Clicker  
 Pulse period (s) (0 for constant; enter e.g. 0.1 for 10 Hz): 0.333  
 Stimulus 2:  Houselight  Centre light  Light over lever  Light on opposite side  Tone  Clicker  
 Pulse period (s) (0 for constant; enter e.g. 0.1 for 10 Hz): 0.1  
 Reinforcer:  Pump - Infusion duration (s): 7.28  
 Dipper - # dips: 1 Dip time (s): 5 Inter-dip time (s): 1  Down at rest  
 Pellets - # pellets: 1 Pulse length (ms): 45 Time between pellets (s): 0.5

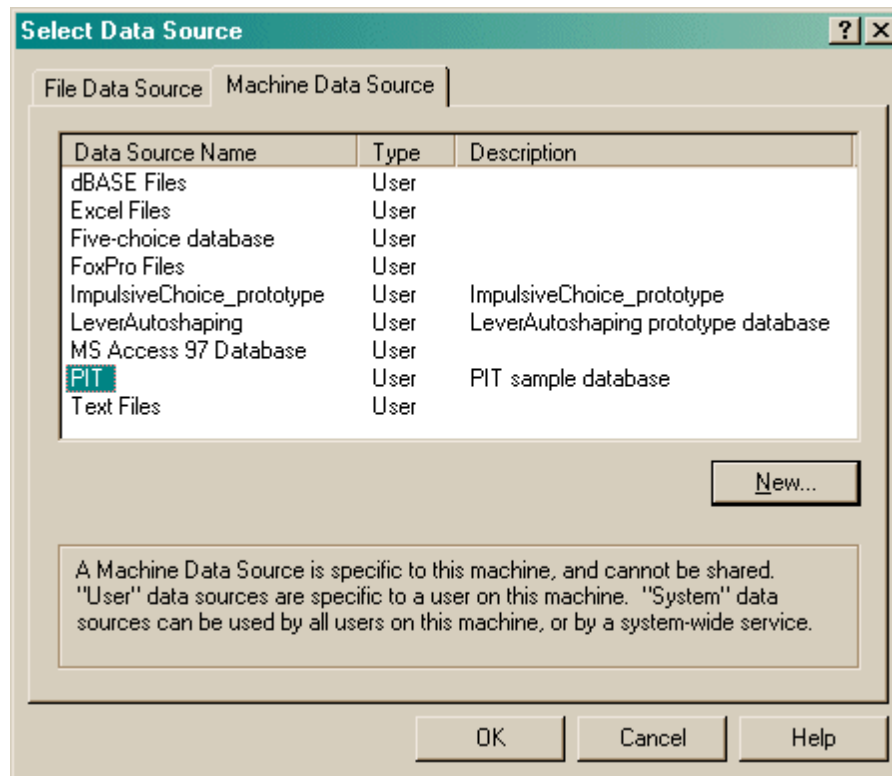
**Schedule**

INSTRUMENTAL training For food, typical RI parameter on subsequent days is 2, 15, 30, 60, 60, 60, 60 (session length 30 min).  EXTINCTION  
 Parameter of random interval schedule (s): 60  Retract lever during timeouts?  
 Time out after reinforcement is complete (s): 20 A 30-min extinction session is given after Pavlovian training, just before the transfer test. Max no. of reinforcers (0 = no limit): 0  
 Session length (min): 30  Use FI (not RI) schedule

PAVLOVIAN training  Pavlovian phase is really discriminative stimulus SD:(nosepoke -> reinf)  
 Number of ISI-CS cycles: 6 During CS, reinf. delivered on an RT schedule, with parameter (s): 30  
 CS duration (min): 2 Time out between reinforcement and the schedule restarting (s): 20  
 ISI duration (min): from 2 to 4  
 Give NEUTRAL stimulus:  never  on the 2nd of every 4 cycles  on the 5th and 7th of every 8 cycles

TRANSFER TEST  
 CS and ISI duration (min): 2 Number of cycles: 6  
 Each cycle is:  ISI-CS  (ISI, CS, NEUT randomized), with no consecutive identical pair overall  
 ISI-CS-NEUT  ISI-NEUT-ISI-CS

To pick an ODBC database **in advance** of finishing, click *Pick* and you will be offered the ODBC Data Source picker (below). Your choice will be recorded and will apply to this subject from now on (or until you specify a different source).



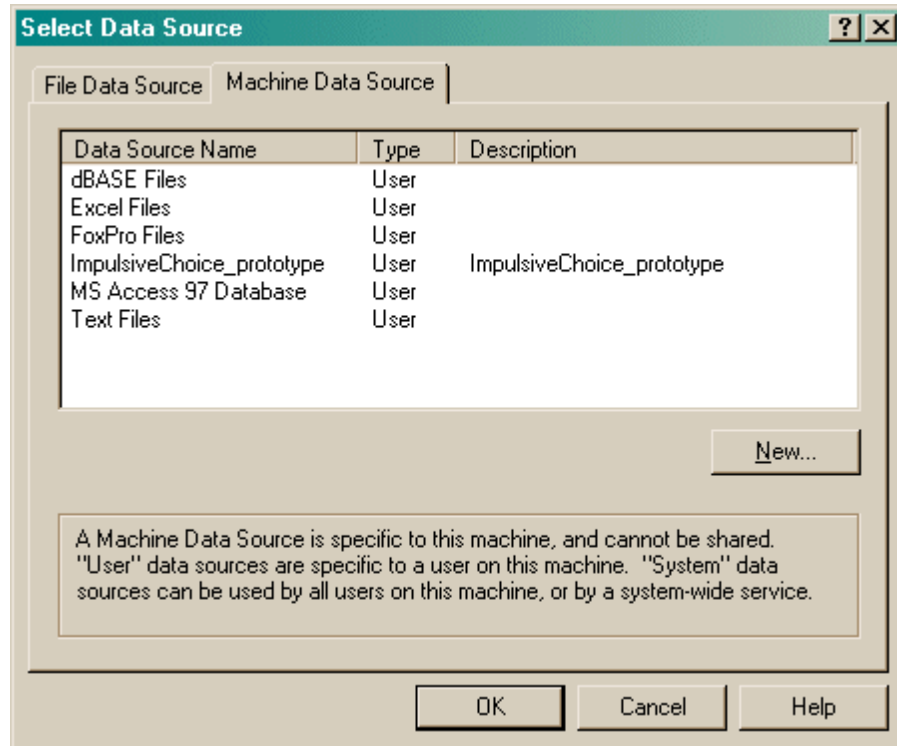
If you don't specify an ODBC data source now, or you delete the value in the "ODBC data source name" box, you'll be asked to choose when the task ends (and that choice will only apply to the session in progress).

[What happens if you can't find an appropriate ODBC source?](#)

## 1.5 Creating a new ODBC source

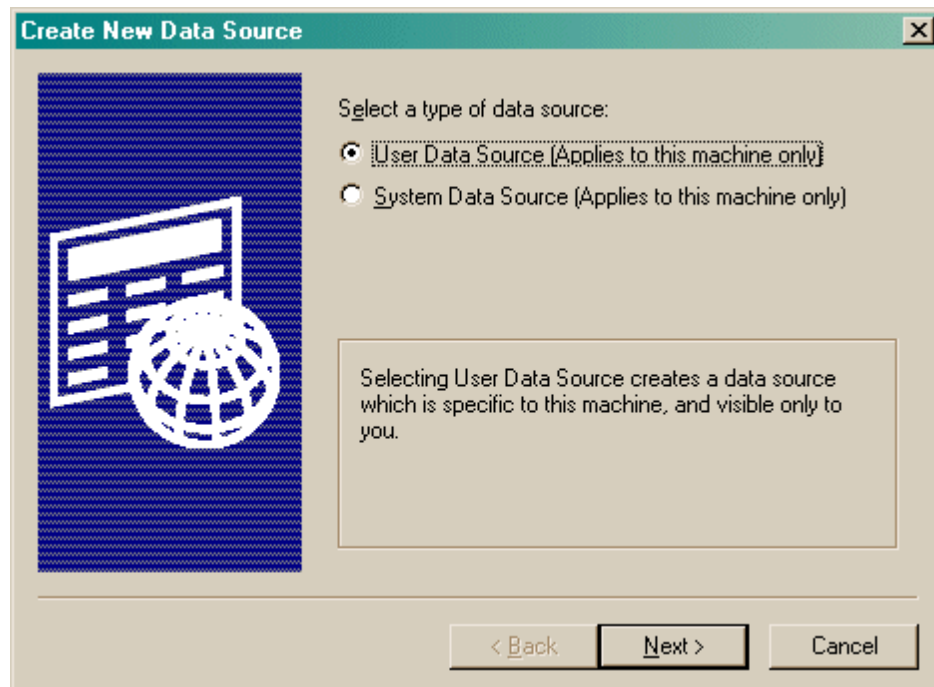
What happens if you can't find an ODBC source? You should configure it via *Control Panel* → *ODBC*. Alternatively, you can set one up "on the fly", as explained here.

Suppose you're looking for a PIT database. But there isn't one...

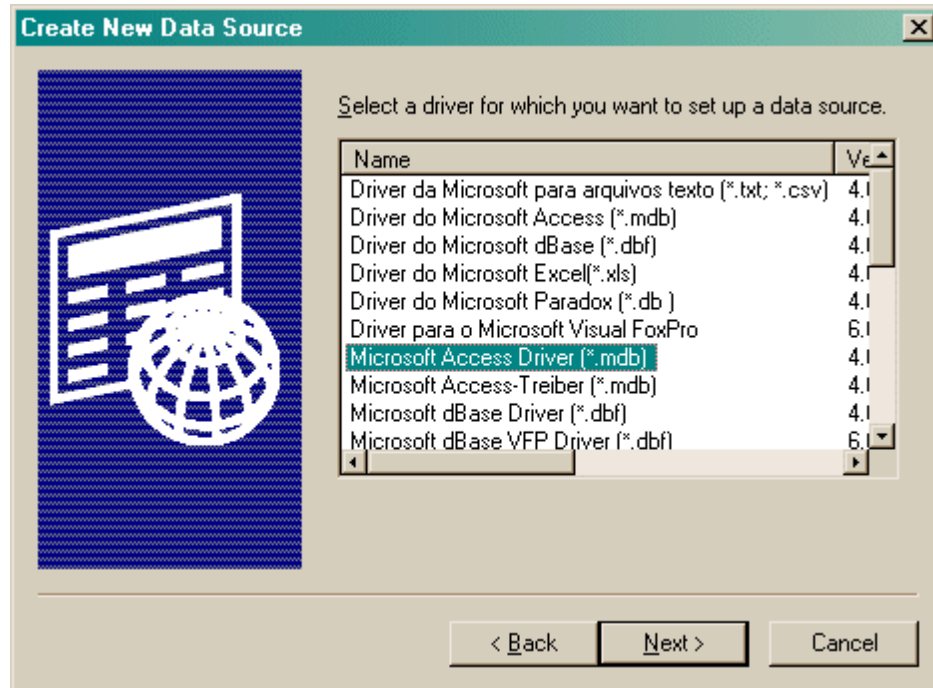


Let's assume that you have **already made a working copy of the prototype database supplied with the task**. How do we go about setting this up as an ODBC data source?

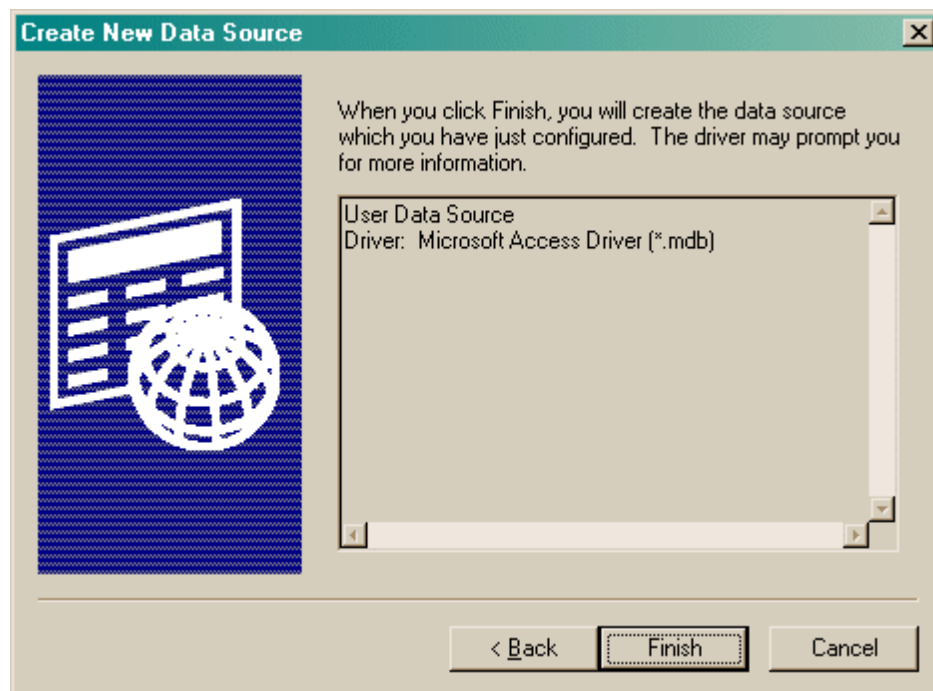
Click New.



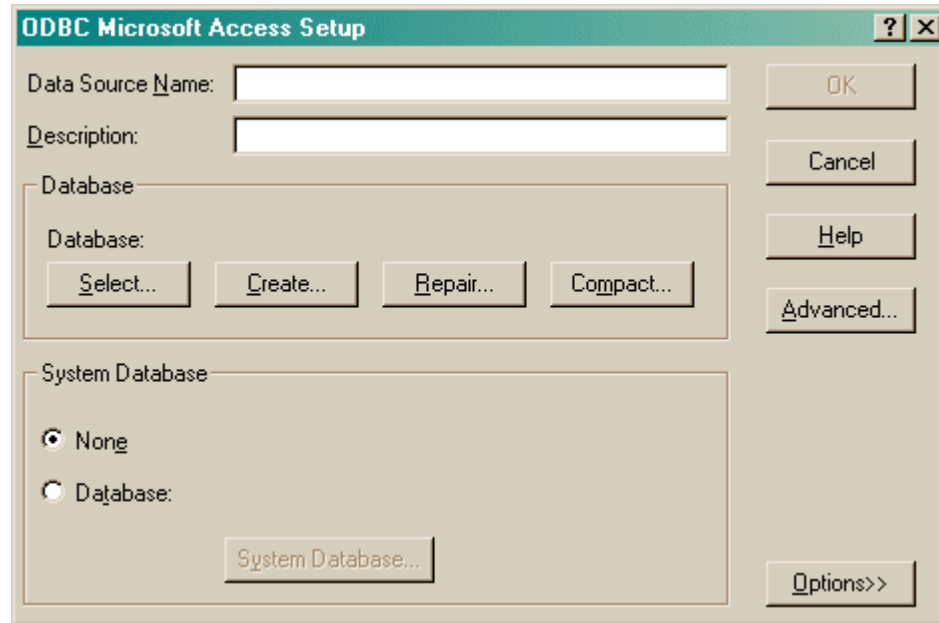
Choose a User or System data source. **User** is probably more sensible. Click Next.



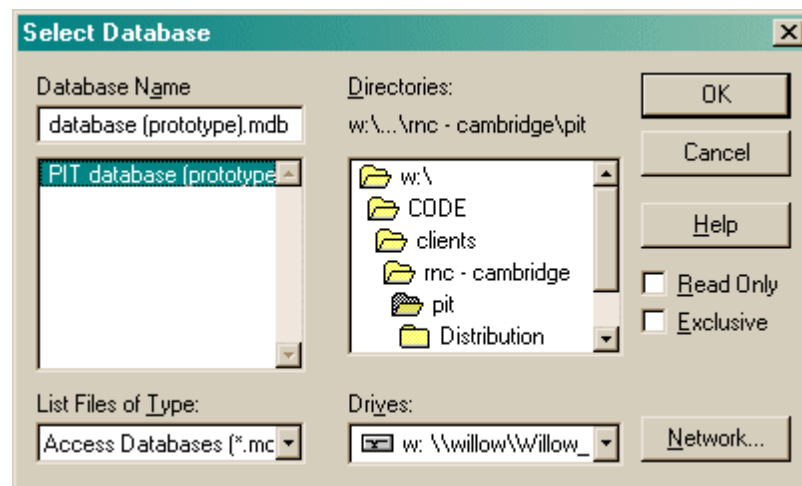
Choose your database driver. Click Next.



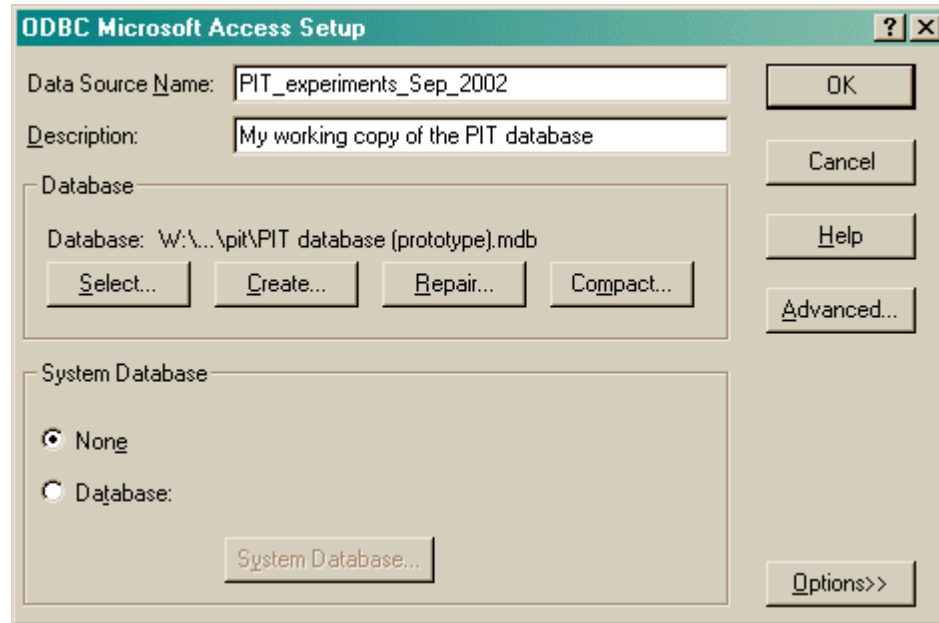
Click Finish.



You should fill in the **Data Source Name (no spaces)** and the **description**, and **Select** a database. When you click Select, this dialogue box appears:



Choose your database here and click OK. Your ODBC data source fields should now all be set up:



Click OK. You will be returned to the ODBC selection screen with your new data source now available.

# Index

## - P -

### PIT

- about 2
- parameters 6
- required devices 2
- using 4