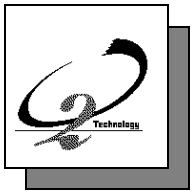

AUTO-DEMO
User Manual

Release 5.0 - April 1998

Who should read this manual

When O_2 is delivered, you receive a demonstration called **auto** with the installation tape. This demonstration uses O_2C, O_2Look, O_2Kit and O_2Graph.

This document explains how to install and use this demonstration.

See O2 Documentation set.

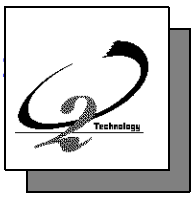

TABLE OF CONTENTS

This manual is divided into the following chapters:

- 1 - Introduction
- 2 - Installation - explains how to install and run the auto demonstration.
- 3 - Scenario - explains how to install and run the auto demonstration.

TABLE OF CONTENTS

1 Installation

AUTO DEMONSTRATION

This chapter introduces the O_2 System through a demonstration.

It contains the following sections:

- System Overview
- The O2 Auto demonstration

1 Installation

1.1 System Overview

The system architecture of O_2 is illustrated in Figure 1.1.

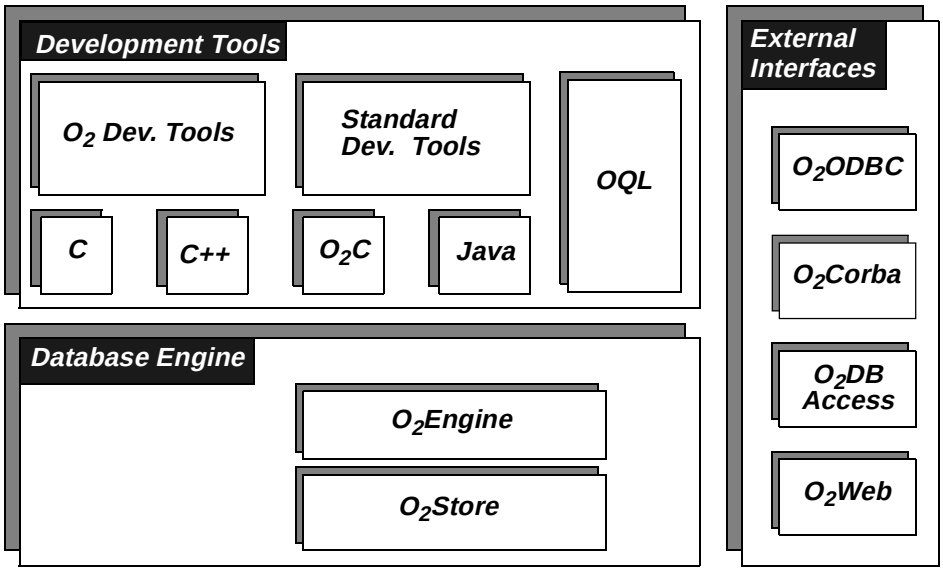

Figure 1.1: O_2 System Architecture

The O_2 system can be viewed as consisting of three components. The Database Engine provides all the features of a Database system and an object-oriented system. This engine is accessed with Development Tools, such as various programming languages, O_2 development tools and any standard development tool. Numerous External Interfaces are provided. All encompassing, O_2 is a versatile, portable, distributed, high-performance dynamic object-oriented database system.

Database Engine:

- O_2Store — The database management system provides low level facilities, through O_2Store API, to access and manage a database: disk volumes, files, records, indices and transactions.
- O_2Engine — The object database engine provides direct control of schemas, classes, objects and transactions, through O_2Engine API. It provides full text indexing and search capabilities with O_2Search and spatial indexing and retrieval capabilities with O_2Spatial. It includes a Notification manager for informing other clients connected to the same O_2 server that an event has occurred, a Version manager for handling multiple object versions and a Replication API for synchronizing multiple copies of an O_2 system.

Programming Languages:

O_2 objects may be created and managed using the following programming languages, utilizing all the features available with O_2 (persistence, collection management, transaction management, OQL queries, etc.)

- C — O_2 functions can be invoked by C programs.
- C++ — ODMG compliant C++ binding.
- Java — ODMG compliant Java binding.
- O_2C — A powerful and elegant object-oriented fourth generation language specialized for easy development of object database applications.
- OQL — ODMG standard, easy-to-use SQL-like object query language with special features for dealing with complex O_2 objects and methods.

O_2 Development Tools:

- O_2Graph — Create, modify and edit any type of object graph.
- O_2Look — Design and develop graphical user interfaces, provides interactive manipulation of complex and multimedia objects.
- O_2Kit — Library of predefined classes and methods for faster development of user applications.
- O_2Tools — Complete graphical programming environment to design and develop O_2 database applications.

Standard Development Tools:

All standard programming languages can be used with standard environments (e.g. Visual C++, Sun Sparcworks).

External Interfaces:

- O_2Corba — Create an O_2/Orbix server to access an O_2 database with CORBA.
- O_2DBAccess — Connect O_2 applications to relational databases on remote hosts and invoke SQL statements.
- O_2ODBC — Connect remote ODBC client applications to O_2 databases.
- O_2Web — Create an O_2 World Wide Web server to access an O_2 database through the internet network.

1.2 The O_2 Auto Demonstration

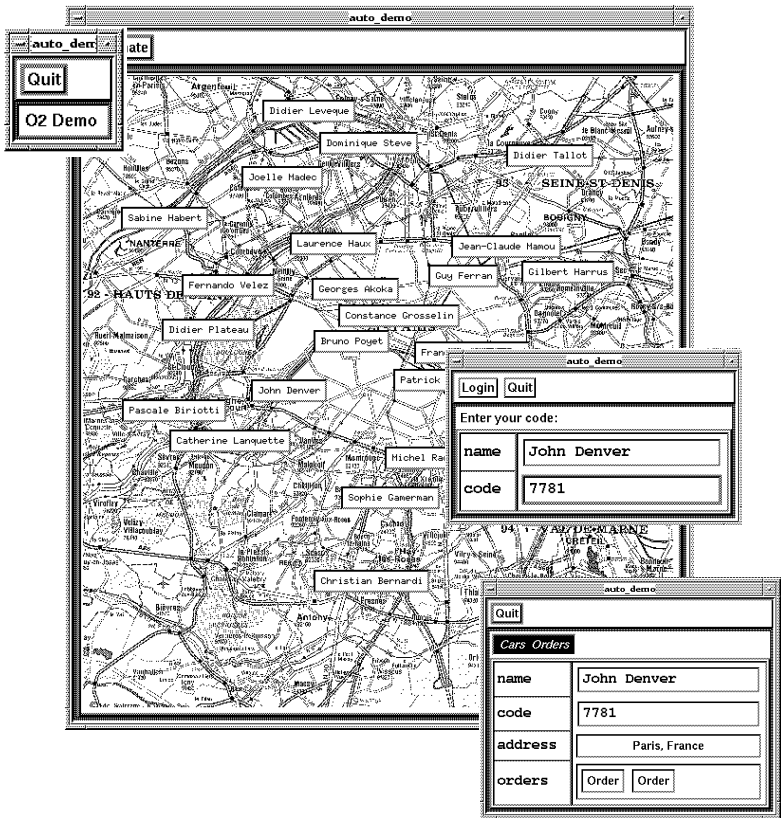

Figure 2: The O_2 Auto demonstration

2 Installation

This chapter describes all the steps you need to carry out in order to install and use the O_2 demonstration **`auto`**. In most of the steps given below, the explanation is followed by the actual command.

2 Installation

Unload the O_2 installation Cd-Rom / tape into a directory called O2HOME.

It is assumed that an **O2HOME** environment variable specifying the access path to this directory now exists.

In **O2HOME**, you find different files and directories.

The following directories are important:

- **bin** directory containing the O_2 binary files
- **look** directory containing O_2Look files
- **demo** directory containing demonstrations

Position yourself in **O2HOME**:

```
> cd $O2HOME
```

Add the **O2HOME/bin** directory to the path in each working window:

```
> set path = ($O2HOME/bin $path)
```

In **O2HOME**, create a directory called **o2vol** to contain all the different O_2 catalogues.

```
> mkdir o2vol
```

In **O2HOME**, now create the systems file needed to define the different O_2 systems.

You should refer to the *O_2C Reference manual* for more details on this file and the organization of O_2 catalogues.

In the **systems** file, insert a line corresponding to the description of the system **demo**. Use the exact syntax of the following command on a single line:

```
> echo demo:`hostname`:$O2HOME/o2vol::$O2HOME/o2vol::$O2HOME/o2vol::>>systems
```

Create the symbolic links that position the resource files of the demonstration for O_2Look:

```
> ln -s $O2HOME/demo/auto/resources/english/auto_demo look/english

> ln -s $O2HOME/demo/auto/resources/french/auto_demo look/french
```

Initialize the system **demo:**

```
> o2init -system demo
```

Launch an O_2server. If you do this in another window, make sure that you add the directory **O2HOME/bin** to the **path.**

```
> o2server -system demo
```

To specify which language you want the demonstration to be in, **english or french,** position the environment variable **O2RESOURCE** to one of the following:

```
> setenv O2RESOURCE english
```

or

```
> setenv O2RESOURCE french
```

Position yourself in the directory **demo** and launch an O_2 client:

```
> cd demo/auto

> o2 -system demo
```

In the client window, load the demonstration **auto:**

```
#"load"

^D
```

Note

If you are using SunOs, do not worry if you receive the following error message:

```
ERROR 20048: Not available on this platform
```

Now you can launch the demonstration **auto** in the client window:

```
run application demo

^D
```

How to use the demonstration is described in Chapter 3 : Scenario.

To quit the demonstration, click (with left mouse button) on the "Quit" button of the O_2 application icon shown in Figure 3 below.

Figure 3: Application icon

To now leave O_2, type in the client window:

```
quit

^D
```

Finally kill the server.

```
o2shutdown -system demo
```

2 Installation

3 Scenario

This chapter goes through a scenario of the demonstration **`auto`** step-by-step.

Note

When you read below to click on a particular part of the demonstration, you should always use the **left** mouse button **unless** it is otherwise stated in the text to use the **right** button.

3 Scenario

When you launch the demonstration **`auto`,** as explained in the previous section, an application icon is displayed as shown in Figure 4.

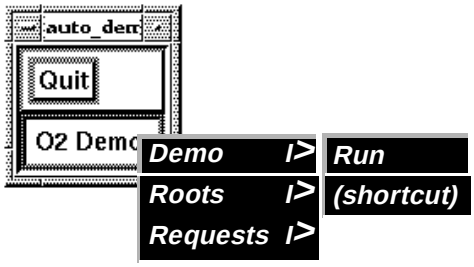

Figure 4: Application icon and menus

Now click on the words **O2 Demo** using the **right** mouse button and a pull down menu appears.

Maintaining the **right** mouse button over the ***Demo*** item another menu appears. Release the button over the item ***Run*** of this menu in order to start up this program.

A map of Paris now appears covered with various name tags.

Click using the **right** mouse button on the name tag **John Denver** (**A**) in the center of the map and select the method **`Display`**.

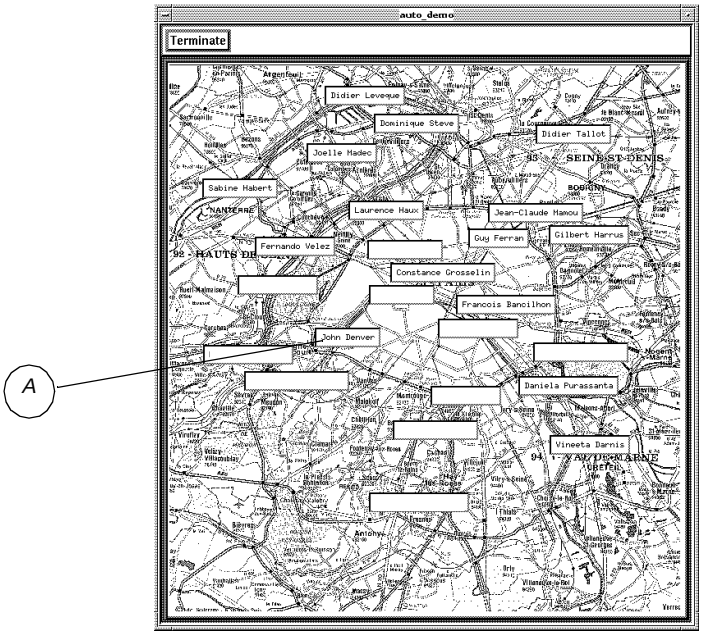

Figure 5: Paris map

A login window now appears as shown in the figure below.

Type in the code field the code **7781** and click on **Login**.

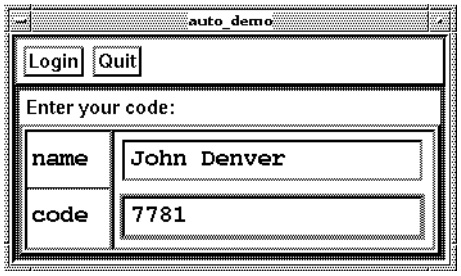

Figure 6: Login window

A window now appears containing information about John Denver including name,

code, address, and orders.

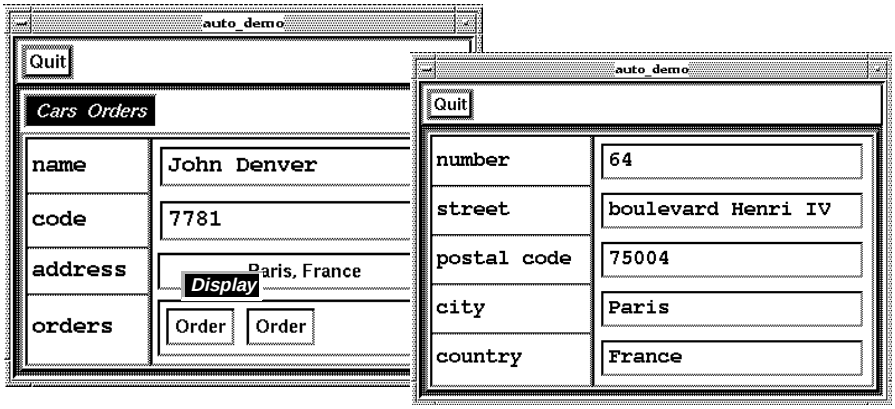

Figure 7: Information boxes

If you now click in the address section of this window using the **right** mouse button, as shown in Figure 7 above, you can select ***Display*** which then gives you details of the address.

Now go back to the John Denver window and select the method **`Display`** in the menu **`Cars`**.

A selection box now appears in which you can choose the set of cars you want to look at: **`All`**, **`France`**. **`Germany`**, **`USA`**. Click on your choice and then on OK.

Figure 8: Country selection

Note

All data concerning French cars is in French and all other data is in English.

All the cars corresponding to the country chosen are displayed - in this case Germany.

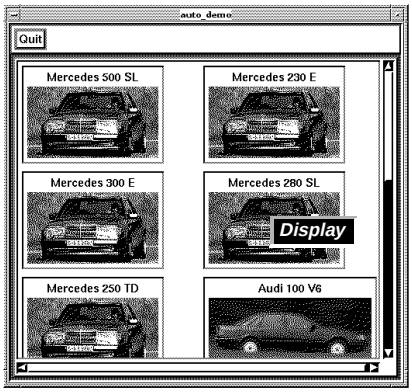

Figure 9: Display of German cars

Click on one of the cars displayed using the **right** mouse button, and select the method **`Display`**. The presentation corresponding to the car chosen is now displayed.

Note

Complete information is only available for the Renault, Mercedes and BMW cars. There is no section cut-out for the Corvette, Audi and Venturi.

This presentation contains technical data, text and two color photos and is shown in Figure 10 below.

Click on the **car manufacturer** section and select the method **`Display`** - you see all the cars produced by that manufacturer in a separate window.

Click on **Quit** to close this window.

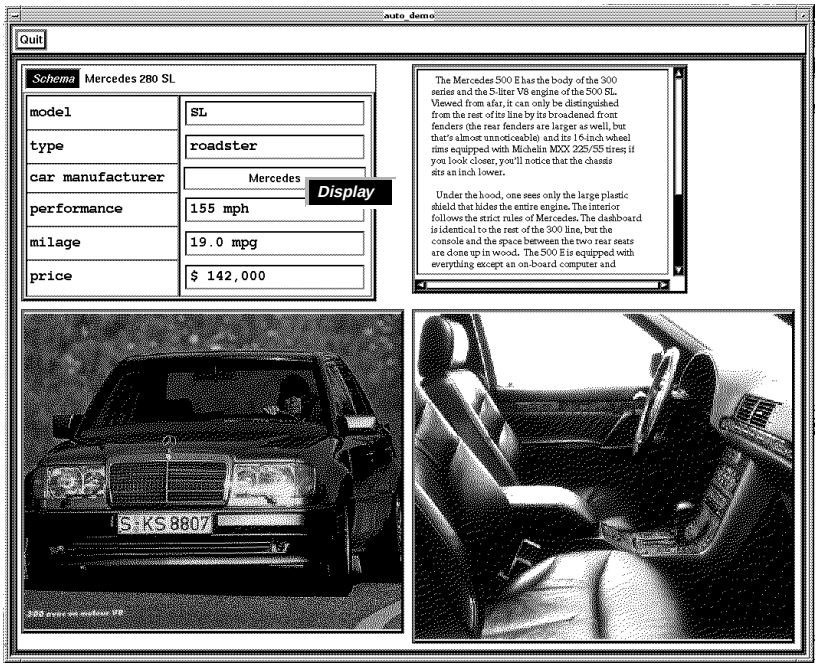

Figure 10: Car information

Reposition yourself in the car presentation window and now select the method **`Display`** in the menu **Schema**.

The section cut-out of the car is displayed. When you click on one of the buttons **`Engine`**, **`Transmission`**, or **`Body`**, the corresponding flags appear on the

diagram.

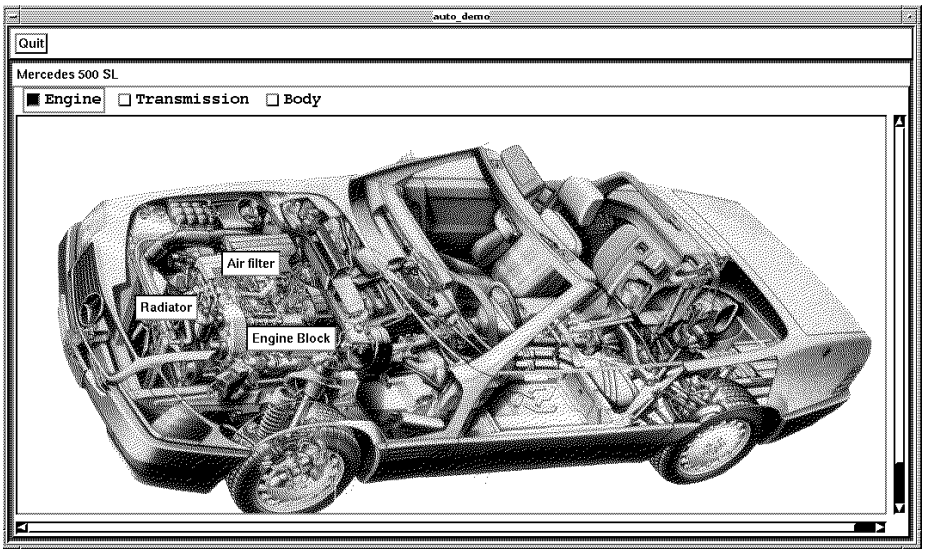

Figure 11: Car cut-out

Click on the button **`Engine.`** The various flags are now displayed. Now click on the Air Filter flag using the **right** mouse button and chose the method **`Schema`**.

A diagram of the Air Filter is displayed and if you click on the Button **`Air Filter`** you see the flags representing the various sub parts of the Filter.

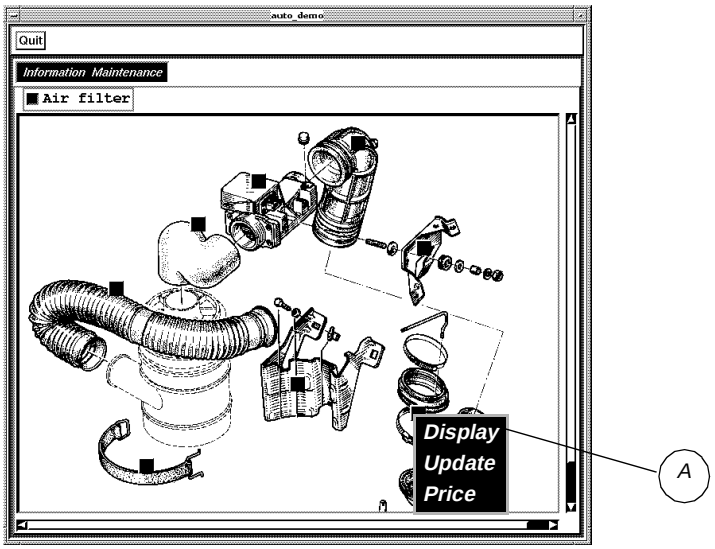

Figure 12: Air filter diagram

If you now click on a subpart flag using the **right** mouse button you see a pull down menu containing the methods **`Display`**, **`Update`** or **`Price`**.

Click on the square on the bottom right hand side using the **right** mouse button and select the method **`Display`** as in (**A**) in Figure 12.

The description of the Air Filter intake assembly is displayed.

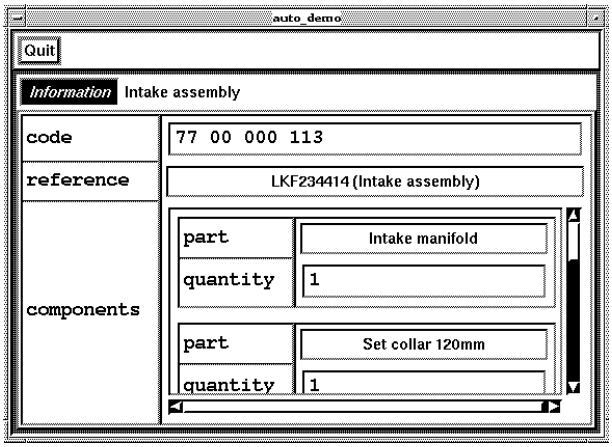

Figure 13: Air filter intake assembly description

Now go back to the Air Filter illustration and click on the ***Information*** menu.

From this menu you can select the methods **`Description`** and **`Price`** to obtain further information about the Air Filter.

If you have the option O_2Graph, you can select from the Air Filter screen the method **`Structure`** to obtain the graph of the Air Filter components.

This is shown in Figure 14

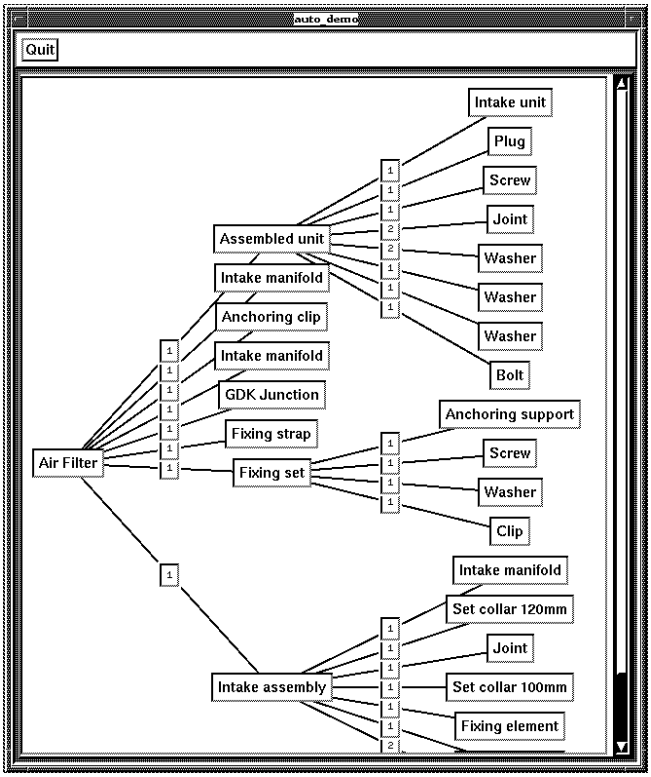

Figure 14: Graph

Before going on with the demonstration, you should now close all windows, by clicking on quit in each presentation, **except** the windows shown in Figure 15, concerning the map of Paris, **`John Denver`** chosen at the beginning, the components graph and/ or the intake assembly components window.

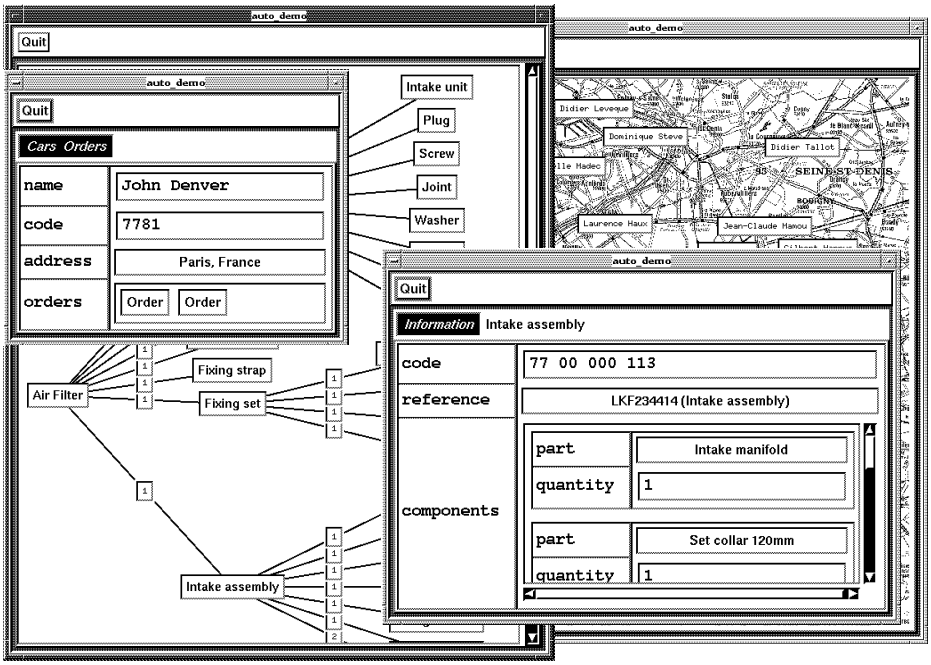

Figure 15: Windows you need

If you now return to the John Denver presentation, select the method **Add** in the menu ***Orders***.

The presentation of an order is displayed.

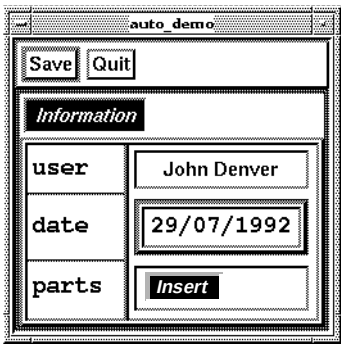

Figure 16: Presentation

Click using the right button in the field **parts,** and select the method **Insert**. A new tuple now appears in the **parts** field - you must enlarge the window to see it clearly.

Enter a quantity. To enter a part, simply select, using the middle mouse button, a flag in the Air Filter Structure graph or a part in the intake assembly components and drag and drop on the blackened square in order to copy the value.

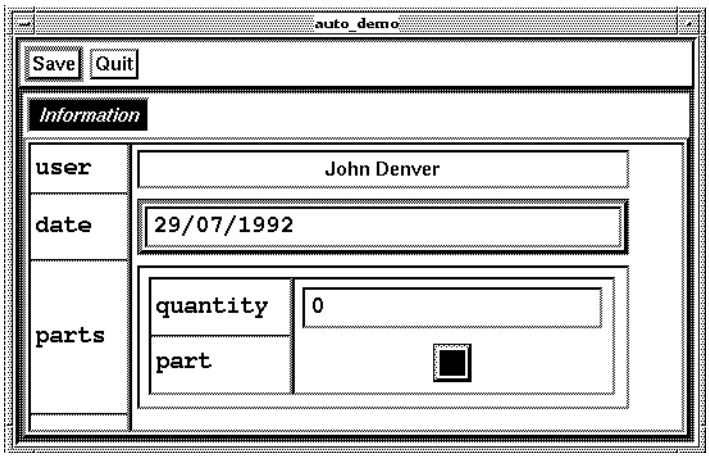

Figure 17: Parts window

You can renew this operation and then on the order presentation select the method **Price** from the ***Information*** menu.

If you have LaTeX on your machine click on the method **Letter**. You now see a preview of a letter in LaTeX. Type q in this window to exit.

You leave the demonstration by clicking on the **Terminate** button on the map of Paris **(*A*)** in Figure 18.

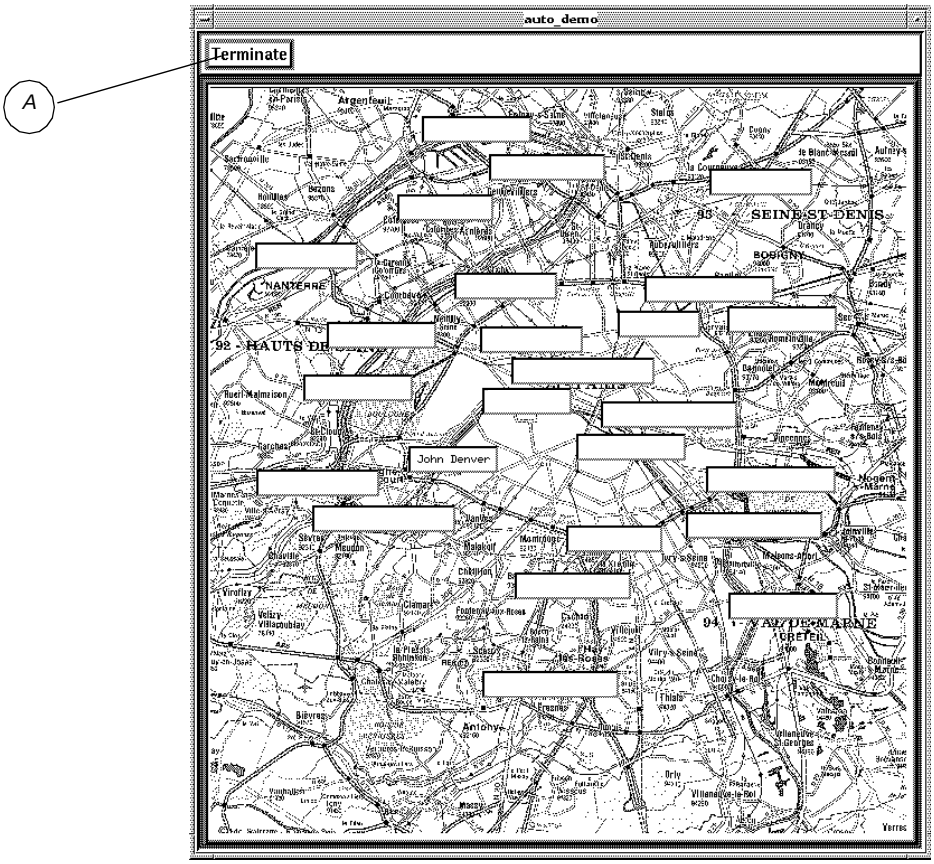

Figure 18: End of demonstration scenario