

<b>CARS USER TUTORIAL</b>	<b>2</b>
Introduction	2
<b>LOGIN SCREEN</b>	<b>3</b>
<b>BUILDING A CARS STRUCTURE</b>	<b>4</b>
<b>PROPERTY FIELDS</b>	<b>5</b>
Table 1: Property Field Types	5
<b>PROPERTY DEFINITION SCREEN</b>	<b>5</b>
<b>ADDING A PROPERTY</b>	<b>6</b>
<b>ADDING DEFAULTS</b>	<b>7</b>
<b>ADDING KEYWORDS, MEMES AND ALIASES</b>	<b>7</b>
<b>DELETING A PROPERTY</b>	<b>8</b>
<b>INHERITANCE</b>	<b>8</b>
<b>CARS OBJECTS</b>	<b>9</b>
Table 2: CARS Object Definitions	9
<b>PUBLISHING CARS DOC</b>	<b>10</b>
<b>UPLOADING TO CARS FILE GROUP</b>	<b>10</b>
<b>MODIFYING CARS OBJECT PROPERTIES</b>	<b>11</b>
<b>SEARCHES</b>	<b>11</b>

# CARS User Tutorial

## Introduction

The function of the Corporate Archive and Retrieval System (CARS) is, to put it simply, an information retrieval system. It addresses the needs of a corporate organisation in collating and, more importantly, in retrieving the masses of information that are generated on a daily basis. This information can take many forms and within CARS we have identified three specific classes of objects that can be stored and searched within the system. An object, in the context of CARS, is an item that can be stored and retrieved in the CARS system.

These objects are not an exhaustive list and will likely expand with the needs and requirements of the organisation. With these it is possible to store emails, faxes, PDF documents, word files, images, zip files etc.

Storing this data in a central location is not the only goal behind CARS but its ability to retrieve this information using a set of property fields that can be 'applied' to each object stored in a folder. These properties give the user the necessary tools to filter objects pertaining to certain conditions (see later sections on *Searches*)

This tutorial looks at the functionality of CARS and how users can create and search within the CARS environment.

It assumed that the user of this tutorial has full access to the system.

## Login Screen

1. Users can login to the system by opening a web browser such as Netscape and entering the following line in the 'Location' field:



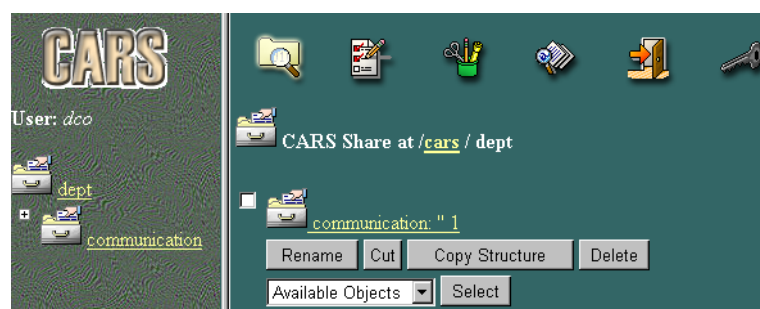
2. The following splash screen will appear:



3. Next enter the share name for which you have access. Your Departmental Information Manager-Web Intelligent Technician (*DIM-WIT*) would provide you with these details. In this example we have access to the *dept* share. Then select GO

**Note: Share names are case sensitive, so enter capital letters when necessary and spaces are not accepted.**

2. You will be presented with a dialog box requesting username and password. Enter your usual email username and password
3. On successfully entering the CARS system you will be presented with a screen similar to the following:



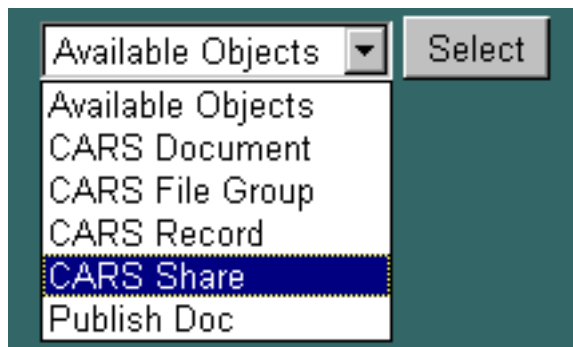
## Building a CARS Structure

For users to successfully store and retrieve data on the system an organised structure befitting the data must be in place. This is the most difficult part of the entire operation of CARS. It should be performed by a person who has knowledge of the data expected to be stored, how it will be accessed and by whom. With a good design it should be obvious to users where certain data is likely to be stored and how to search within certain folders.

The structure of CARS is similar to folders found on operating systems like Linux and Windows where they hold data files and documents relating to a similar content. These folders are referred to as *Shares* in the CARS environment. Shares provide the builder of a CARS system with the ability to control who has access to the information stored within it e.g. user *dco* will have only read access. It also provides the capability of attaching *property fields* to every object stored in that folder e.g. every object will have a title, keyword list and upload date.

Lets begin:

1. From the drop down menu list in the web browser, select **CARS Share** and press the *Select* button:



2. You will then be asked for the Id, Title of the share and the Revision. For the moment ignore the Revision and enter a valid Id for the share e.g. communication. Press *Add* and if no other object exists with that name it will be created. Remember no spaces!
3. A share will now have been created with the name *communication*. Next create additional shares that follow this structure. They are created in the same manner as detailed above.

### *dept*

- communication
- companies
  - BMW
    - in
    - out
- contact

Now that we have a skeleton structure on which to work we will examine shares in more detail.

## Property Fields

As previously described *property fields* (also referred to in this document as either *properties*, *fields* or *attributes*) can be applied to shares. These properties are then inherited by each object within that share. This allows objects to be searched for under certain criteria. Table 1 describes the type of properties that can be applied.

Property Field Type	Details
<i>string</i>	a single line of text providing a concise label for a given property
<i>date</i>	property containing a date
<i>text</i>	a multi-line text field allowing a detailed description to be inserted e.g. an abstract
<i>lines</i>	used mainly to represent a set of keywords/phrases. A more complex search pattern can be implemented with this type than with text searches.
<i>meme</i>	a special type allowing the selection of one element from a group of similar ideas e.g.
<i>int</i>	an integer value
<i>long</i>	a very big integer
<i>float</i>	a number containing a decimal point
<i>boolean</i>	True/False option
<i>email</i>	the valid email address that will start an email application when it is click on
<i>inform</i>	a list of user ids to be informed by email when additions/modifications of objects within and below this share are performed
<i>objref</i>	a link to other objects within CARS

Table 1: Property Field Types

## Property Definition Screen

1. Click on the navigation frame on the browser and click on *companies*, then click on the following icon (this is the property definition function):



2. You will be presented with a screen that will enable you to insert additional properties. From here you can also define default values that all objects will inherit automatically. The following image details the important areas on the screen.

**Define Properties:**

Label	Default	Seq	Type
Title	<input type="text"/>	1	string

**Add Property:**

Name  Type

Label  Seq

### Define Properties:

- *Label* - defines the viewable name of the property
- *Default* - place here the default value that will be automatically assigned to objects in the share
- *Seq* - the sequence number lists the property in ascending order. Values can contain decimal places e.g. 5.8
- *Type* - the type of the property as listed in Table 1

### Add Property:

- *Name* - the unique id of the property. It will not be viewable.
- *Label* - defines the viewable name of the property
- *Type* - select from the list the type of the property
- *Seq* - the sequence number lists the property in ascending order. Values can contain decimal places e.g. 5.8

The other buttons will be described in a later section.

## Adding a Property

1. Enter the following properties in *companies* using the *Add Property* section:

Property Name	Property Label	Type	Sequence
date	Date	Date	2
kws	Keywords	lines	3
contact	Contact	objref	4
info	Info	text	5
via	Via	meme	6
inform	Inform	inform	99

2. Then enter for the *contact* share:

Property Name	Property Label	Type	Sequence
email	Email	email	2
phone	Telephone	string	3
dept	Department	string	4
available	Available	boolean	5

## Adding Defaults

1. Place the following line in the *Contact* property:



Contact /cars/dept/communication/contact 4.0 objref

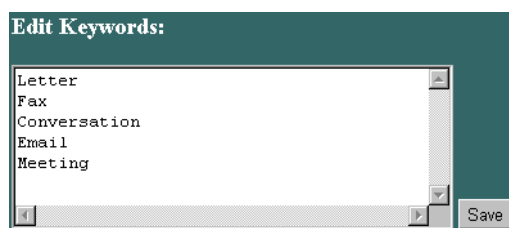
2. And in the *Inform* property insert your user id e.g. dco in this case



Inform dco 99.0 inform

## Adding Keywords, Memes and Aliases

1. Press the *Edit Keywords* button and enter the following:

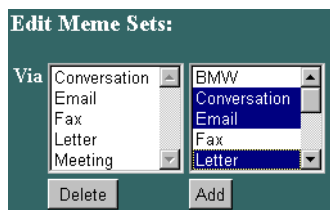


**Edit Keywords:**

Letter  
Fax  
Conversation  
Email  
Meeting

Save

2. Save the data before going back to the edit screen again.
3. Press *Edit Meme Sets* and select from the *add* list the following; *conversation*, *email*, *fax*, *letter*, *meeting*



**Edit Meme Sets:**

Via Conversation BMW  
Email Conversation  
Fax Email  
Letter Fax  
Meeting Letter

Delete Add

4. Returning back to the edit properties screen press the *Edit Aliases* button and enter:

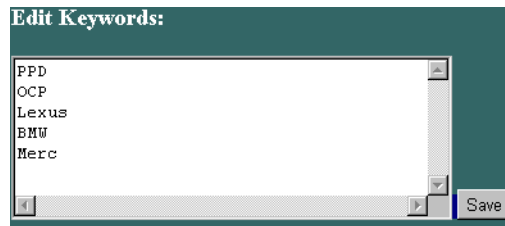


**Edit Aliases:**

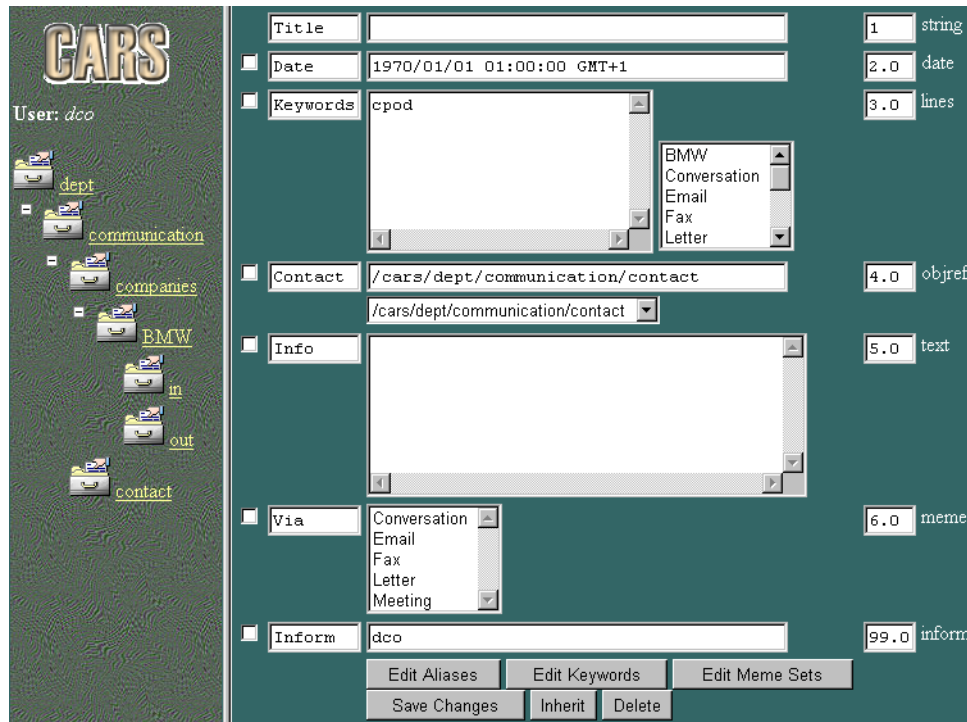
akse  
cpod

Add Alias

5. All that is left to do now is place in the default field of *Keywords* the value *cpod* and save changes.
6. Select the *communication* share and press the *Edit Keywords* button in the property definition screen. Then enter the following data and save.



7. Finally going back to the *companies* share and property definition screen you should have something similar to the following:



## Deleting a Property

Not all properties are deletable. When you first create a property and wish to delete it simply mark the property with a tick and press the *Delete* button. The selection field is a little white box to the left of the property e.g.



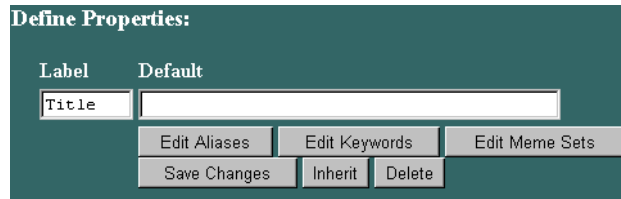
If this box is unavailable then the property has been inherited and cannot be removed at this level (see *inheritance* next section)

## Inheritance

The properties defined in these shares have effect if we place objects only at this level, however if we have sub-shares, that is to say if we have shares within shares, an organisational hierarchy is created and in order for these properties to be available in lower shares we must explicitly inform these sub-shares that that is our intent. The following step must be applied if objects are to have these newly defined properties at lower levels:



1. In all sub-shares below *companies* ( e.g. *BMW, in, out*), go to the property definition screen and press the *Inherit* button:



2. All properties of the parent share will be inherited, along with any default settings. Note that there will be no opportunity to delete these properties. The only way to remove them is to press the *Uninherit* button. You may do this numerous times without affecting the objects contained within this share and lower.

## CARS Objects

There are three main types of cars objects that were mentioned in previous sections. These are:

CARS Object	Description
<i>CARS Doc</i>	<p>A document object published in CARS either via an application, such as Word etc., using the CARS converter (similar to the PDF converter) or via a scanner for paper documents.</p> <p>The pages of the document are converted into images and cannot be modified by the user. Each page is viewable interactively via the web browser.</p> <p>In addition it is possible to attach the original document to the object for print purposes. However any modifications to the object stored in CARS is not possible unless explicit security rights have been allocated to an individual.</p>
<i>CARS File Group</i>	<p>An object that stores a collection of files that can be downloaded and viewed locally on a users' machine. Certain files can be viewed via the web browser if the appropriate plugin is applied to the browser e.g. PDF viewing, Word files etc.</p>
<i>CARS Record</i>	<p>A record is the simplest form of object in CARS. It does not allow any files or document pages to be attached to it. Its function is to enable data to be inserted into property fields which it inherits from the share in which it resides. The analogy is similar to a record in a database, where fields define the type of data that the record will contain.</p>

Table 2: CARS Object Definitions

## Publishing CARS Doc

When working on an electronic document it is possible to print this document to CARS via a CARS converter. This converter works in a manner similar to the PDF converter. You are informed of the success or failure of the document being inserted into CARS.

1. Select the CARSconverter as the printer you wish to send the document to and wait for the appropriate message indicating a success.
2. Next go to the location in CARS, via the web browser, and select *Publish Doc* for the Available Objects list. If you do not have the correct authority to publish a document then you will not have this list available to you.
3. Select the file from the list displayed and give it a unique Id and Title of document.
4. To attach the original file to the object click on the following icon:



5. Then click on the *Browse* button and select the original file in your local hard disk:

**Attach Original Document:**

Current *(None)*

Document

## Uploading to CARS File Group

1. A CARS File Group can contain many different types of files. To attach and delete files select the upload icon:



2. Then using *Browse* select the file from the local file system and *Upload* . The *File Id* field allows you to give the file a new label when stored in CARS.

**Upload File:** **Delete File:**

File

File Id

3. To delete a file just select it from the list on the right and press the *Delete* button

## Modifying CARS Object Properties

Previously we defined property types for the shares *companies*, *BMW*, *in* and *out*. The power of CARS is its ability to apply these properties to every object contained within a share. However, when an object is included into a share the properties that it obtains are initially null, unless default values were defined. To make an effective search of all objects in these shares requires that a certain amount of detailed information be attached to these properties for each object added. This is accomplished in the following procedure:

1. Create an object, e.g. *CARS Record* in the following location:  
*dept/communication/companies/BMW/in*
2. Give it any Id and Title that you wish.
3. Next click on the object that you just created and click on the following icon:



4. You will be presented with the properties that you defined earlier. Fill in the details that you like. The more information that you include in this object will allow you to make more refined searches in the future.
5. Remember to save the changes you made.
6. Create a few other objects they can be of different types. Experiment.

## Searches

Now that a few objects have been introduced to CARS we can try to locate specific objects that we want using searches.