



---

Testament: Test Entity Reservation and Tracking  
Version 1

CML00071-01

---

Code Magus Limited (England reg. no. 4024745)  
Number 6, 69 Woodstock Road  
Oxford, OX2 6EY, United Kingdom  
[www.codemagus.com](http://www.codemagus.com)  
Copyright © 2014 by Code Magus Limited  
All rights reserved



December 15, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Overview</b>	<b>3</b>
<b>3</b>	<b>User Guide</b>	<b>5</b>
3.1	Logging On . . . . .	5
3.2	The Main Menu . . . . .	6
3.3	Maintaining Group Information . . . . .	8
3.4	Maintaining Type Information . . . . .	9
3.5	Working with Reservations . . . . .	11
3.6	Working with Reservation Linked Data . . . . .	13
3.7	Working with Reservation Pool Data . . . . .	15
3.8	System Authorisation . . . . .	18
<b>4</b>	<b>Installation</b>	<b>20</b>
4.1	Apache on Linux using DB2 . . . . .	20
4.1.1	Set the Code Magus Home Directory . . . . .	20
4.1.2	Unpack the Installation Package . . . . .	20
4.1.3	Copy Apache Configuration File . . . . .	20
4.1.4	Create the DB2 Tables . . . . .	20
<b>5</b>	<b>Customisation</b>	<b>22</b>
5.1	Apache on Linux using DB2 . . . . .	22
5.1.1	Configure the Apache Settings . . . . .	22
5.1.2	Bind the DB2 Plan . . . . .	23
5.1.3	Grant DB2 Permissions . . . . .	24
5.1.4	Add a Reservation System Administrator . . . . .	24
<b>A</b>	<b>Apache Configuration File Template</b>	<b>25</b>

---

## List of Figures

1	System Log In . . . . .	5
2	Main Menu . . . . .	6
3	Group Maintenance . . . . .	8
4	Type Maintenance . . . . .	9
5	Working with Reservations . . . . .	11
6	Swapping From a Reservation to View Link Data . . . . .	12
7	Working with Reservation Linked Data . . . . .	13
8	Working with Reservation Linked Data Detail . . . . .	14
9	Searching for Test Data to Use . . . . .	15
10	Specifying Search Criteria for Searching the Pool . . . . .	16
11	Specifying Criteria for Searching the Pool . . . . .	16
12	Working with Pool Data . . . . .	17
13	Maintaining Authorisations . . . . .	18

## 1 Introduction

Code Magus Limited Testament provides a means for users to cooperate and reserve entities for various purposes especially within a quality assurance testing environment. It is a complete test data management system in that it is an aid to comprehensively managing the test data life cycle; enables quick test data generation; and through the assisted query engine easy look up of and access to available test data from a pool of potential data.

An entity may be any key value within the test data domain; examples being (in a financial application) accounts, customers and credit cards. These entities are often a scarce resource and can be easily compromised if more than one initiative processes them. The reservation of entities is not recorded against a person, but a group. This allows the partitioning of the entities on one system for the use of various activities, such as separate testing teams, on that system. All entities can be manipulated as one by referring to the group, for example once testing is complete the entities of a group can be released together.

The main interface for enquiring about and reserving ad-hoc entities is via a web browser, however the majority of reservations can be and are usually done in bulk via the Testament batch utilities. The batch utilities reserve entities and store the underlying data automatically by periodically harvesting information from data files. Linked entries and the associated transaction data can also be loaded into the system. For example a credit card could be loaded from a master file or data base and reserved for a particular use, the linked customer account could also be reserved and loaded, and subsequently all transactions pertaining to that card or customer could be loaded into the system. This provides a comprehensive audit trail of the actions performed on the entity, and holds both the card and customer accounts for the duration of the reservation, or in other words the duration of the test initiative.

A pool of potential candidates for reservation can also be loaded. The web based query engine allows users to find suitable entities and reserve them immediately for their use.

## 2 Overview

The Code Magus Limited Testament system is a web and server based application that allows users to reserve entities for a purpose. They may also view the entities reserved by other users of the system. The entity and its purpose depend on the application system they refer to; again examples are account numbers, plastics (credit card numbers), merchants etc.

Each entity belongs to a group which allows entities with common attributes to be maintained together. Also, each entity conforms to a type in that its name must match the

constraint imposed by that type. Further more a reservation is unique within the entity name and the type.

Reserving an entity on-line is described in section 3.5 on page 11, while maintaining groups and types are described in section 3.3 on page 8 and section 3.4 on page 9 respectively.

The Testament system includes a pool of candidate entities and related/linked data that may be reserved. These entities can be loaded into the pool in bulk via batch utilities and are then available to users to search for, view, reserve and use for testing. This process is described in section 3.7 on page 15.

The Testament system also includes an authorisation component that ensures the integrity of a reserved entity. This feature is described in section 3.8 on page 18.

The steps required to install the Testament system under the Web Servers it supports are described in section 4 on page 20 and the customisation in section 5 on page 22.

The Testament system also enables, through other portals, a non-interactive interface for making entity reservations, which is especially useful for bulk batch processing.

In order to reserve entities, load related data and populate the pool of candidate entities see the related manual CML00072-01: Testament: Batch Utilities Version 1 [2]

## 3 User Guide

The system is accessed via a web browser and the URL for starting it is:

<http://HOST/Reservation/cgi-bin/cmltert>.

where HOST is the host name or Internet protocol address of the server hosting the system. This URL will present the log in page.

### 3.1 Logging On

The log on screen validates a user's credentials and, if valid, allows access the system. The user logging on may also change their password at the same time.

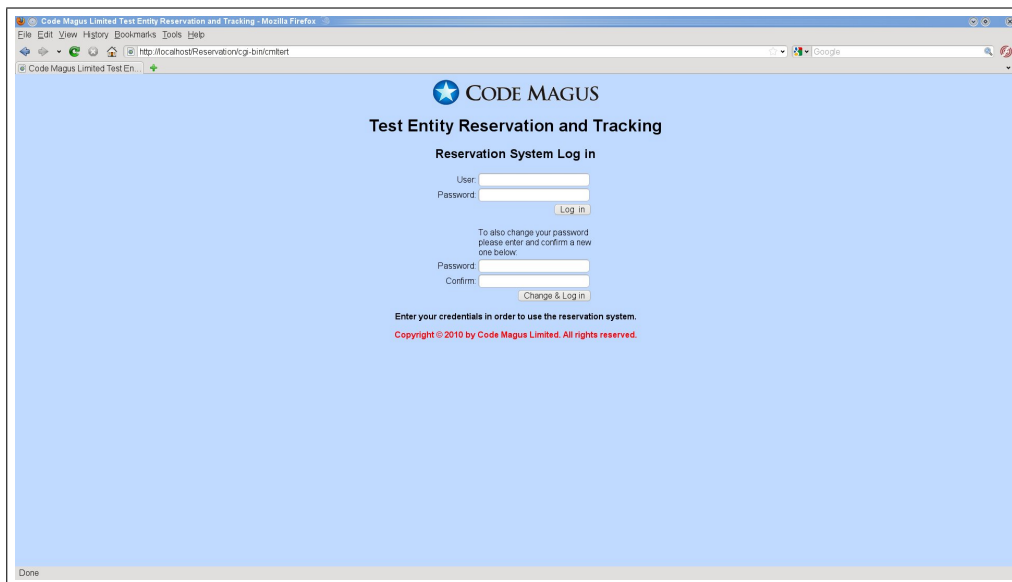


Figure 1: System Log In

Enter a valid user ID and associated password for the system and click the 'Log in' button. Depending on how the system is customised the user ID field may have been filled in with a site specific default value; This value can be over typed in order to change it. If you wish to change your password then enter it twice in the respective fields.

## 3.2 The Main Menu

Once the credentials have been validated and the main menu is displayed a user may, depending on their authorisation attributes, view, add, change or delete reservations, linked data, types and groups. If the user has sufficient authorisation they may also maintain authorisation settings for other users of the system.

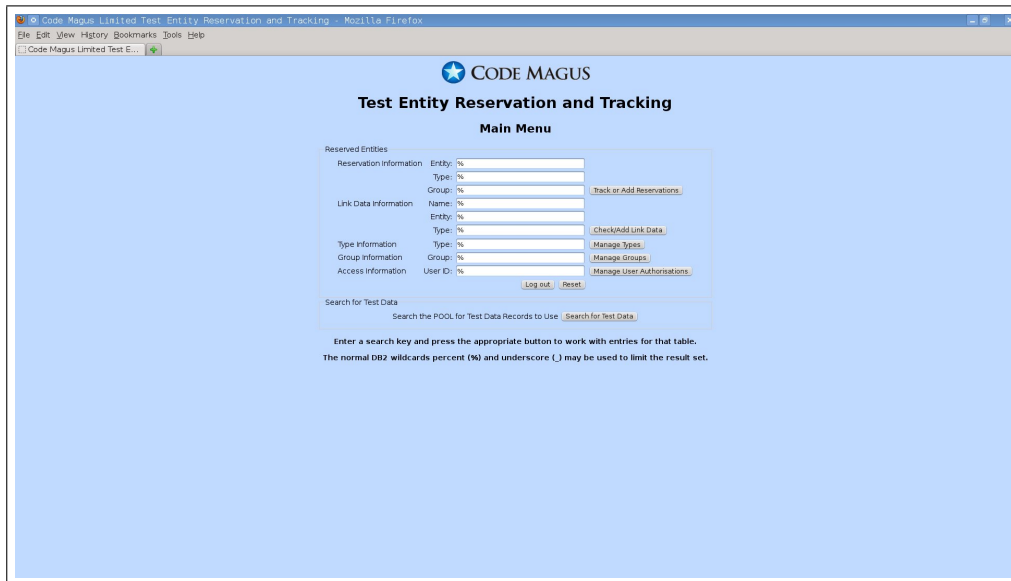


Figure 2: Main Menu

To view or maintain types, groups or authorisations, to view and make reservations or linked data or to work with pool data click on the relevant button. To limit the list of results shown on the maintenance pages enter a selection mask (the default is %) in the box to the left of the button. The selection mask may be an exact value or one that has wild card meta characters embedded in it. Note that for reservations and linked data there are three selection key fields, and it is permissible to use an exact and/or wild card mask in either of the key fields. The wild card meta characters are:

- ‘%’ - percent sign. This matches any number of characters (including zero) in its place.
- ‘\_’ - underscore character. This matches exactly one character in its place.

For example if the types defined on the system are Type1, Type2, type3, Type100 and Type200, then the following masks will match the shown values:

- ‘%’ - This is the default and would select all the types.
- ‘Type%’ - Type1, Type2, Type100 and Type200 (notice that the leading ‘T’ caused type3 to not be matched).
- ‘\_ype\_00’ - Type100 and Type200.

- ‘\_ype\_’ - Type1, Type2 and type3.
- ‘%00’ - Type100 and Type200.
- ‘%1%’ - Type1 and Type100.



### 3.3 Maintaining Group Information

This page displays, up to a configured page maximum, groups that match the mask entered on the main menu. If there are more groups than will fit in the maximum allowed then the buttons 'previous' and 'next' can be used to navigate back and forward through the results.

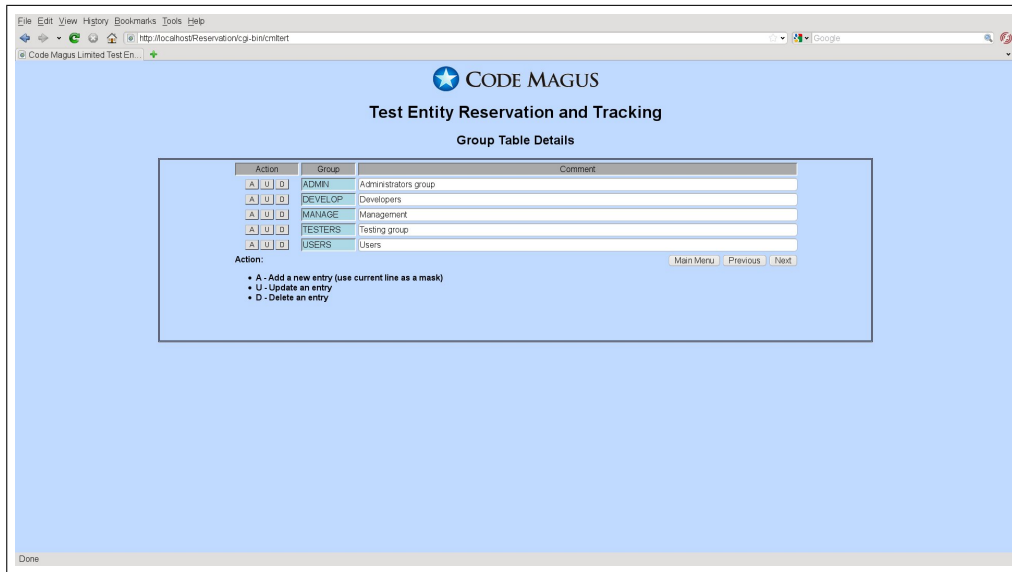


Figure 3: Group Maintenance

Groups are maintained on this page through selecting (by clicking) the applicable mini-button to the left of each row. As the cursor hovers over a button it will turn red. The action characters are:

'D' Delete. This will display a new page asking for confirmation to delete the selected group.

**Note** Deleting a group will cause all entities that belong to that group to also be deleted without any notification of the number deleted.

'U' Update. This will display a new page for updating the group with any values that have been changed. The key fields can not be changed; rather a new group must be added.

'A' Add. This will display a new page for adding groups using the values from the selected group as a template. One or more groups can be added one at a time using the add page and once all the required groups have been added the user can return to the result page or the main menu.

The fields that may be updated are:

- Comment. This is descriptive text that should describe the function of the group.

### 3.4 Maintaining Type Information

This page displays, up to a configured page maximum, types that match the mask entered on the main menu. If there are more types than will fit in the maximum allowed then the buttons 'previous' and 'next' can be used to navigate back and forward through the results.

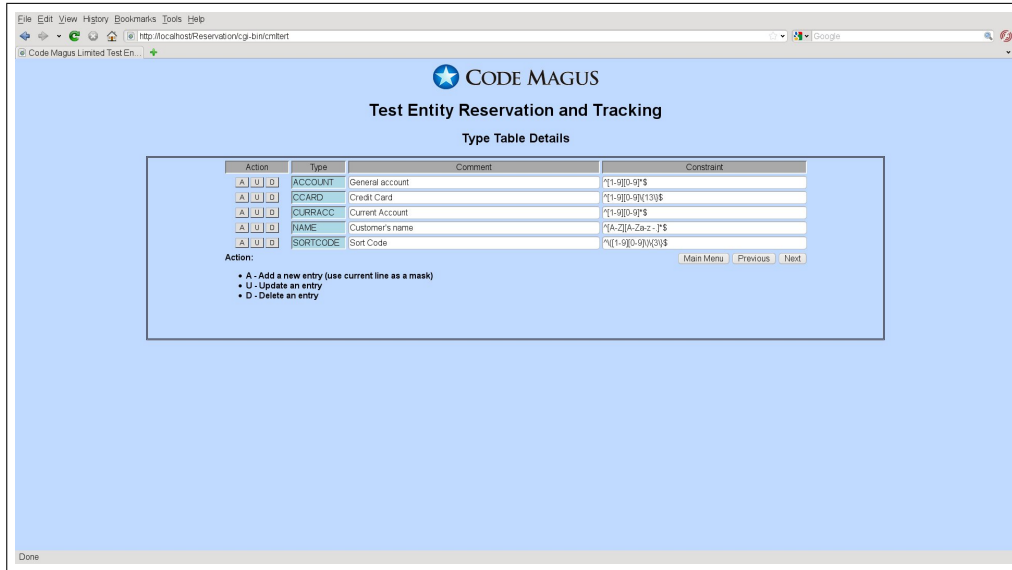


Figure 4: Type Maintenance

Types are maintained on this page through selecting (by clicking) the applicable mini-button to the left of each row. As the cursor hovers over a button it will turn red. The action characters are:

'D' Delete. This will display a new page asking for confirmation to delete the selected type.

**Note** Deleting a type will cause all entities that belong to that type to also be deleted without any notification of the number deleted.

'U' Update. This will display a new page for updating the type with any values that have been changed. The key fields can not be changed; rather a new type must be added.

'A' Add. This will display a new page for adding types using the values from the selected type as a template. One or more types can be added one at a time using the add page and once all the required types have been added the user can return to the result page or the main menu.

The fields that may be updated are:

- Comment. This is descriptive text that should describe the function of the type.

- **Constraint.** This field is written as a regular expression and is used to constrain an entity name when one is reserved.

For example to limit an entity name such as a credit card to start with a digit between 1 and 9 and then allow another 15 digits between 0 and 9 use the following regular expression.

```
^[1-9][0-9]{15}$
```

### 3.5 Working with Reservations

This page displays, up to a configured page maximum, all the reservations that have been made and that match the mask entered on the main menu. If there are more reservations than will fit in the maximum allowed then the buttons 'previous' and 'next' can be used to navigate back and forward through the results.

Any user logged on to the system may make a reservation, delete their own reservations and view all the reservations with out any special access authorisations. In order to update or delete a reservation that does not belong to the logged on user then those specific authorisations need to be given on the access table maintenance page, see section 3.8 on page 18

Action	Entity	Type	Purpose	Group	User ID	Time Stamp
A   U   D   L	377086700000043	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:20:185674
A   U   D   L	377086700000050	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:371507
A   U   D   L	377086700000076	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:431900
A   U   D   L	377086700000084	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:370304
A   U   D   L	377086700000118	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:363762
A   U   D   L	377086700000209	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:304846
A   U   D   L	377086700000308	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:361133
A   U   D   L	377086700000340	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:369789
A   U   D   L	377086700000373	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:20:049008
A   U   D   L	377086700000423	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:432149
A   U   D   L	377086700000464	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:20:128887
A   U   D   L	377086700000621	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:384392
A   U   D   L	377086700000637	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:369295
A   U   D   L	377086700000636	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:365765
A   U   D   L	377086700001110	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:042325
A   U   D   L	377086700001983	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:17:384337
A   U   D   L	377086700002924	TCCARDS	LINKED	TESTINGL	testdata	2011-03-23-18 07:19:367320

Action:  
 • A - Add a new entry (use current line as a mask)  
 • U - Update an entry  
 • D - Delete an entry  
 • L - Display Linked Data Entry

Main Menu | Previous | Next

Figure 5: Working with Reservations

Reservations are maintained on this page through selecting (by clicking) the applicable mini-button to the left of each row. As the cursor hovers over a button it will turn red. The action characters are:

- ‘D’ Delete. This will display a new page asking for confirmation to delete the selected reservation.
- ‘U’ Update. This will display a new page for updating the reservation with any values that have been changed. The key fields and automatically defaulted values can not be changed; rather a new reservation must be added. Values that are automatically defaulted are the user ID that is making the reservation and the time stamp.
- ‘A’ Add. This will display a new page for making a reservation using the values from the selected reservation as a template. One or more reservations can be added one at a time using the add page and once all the required reservations have been added the user can return to the result page or the main menu.

'L' Show Linked data. This option swaps to the link data display page based on the entity and type of the row the selection is made for. See figure 6 on page 12 for an example of selecting 'L' on the first entry of the figure 5 on page 11.

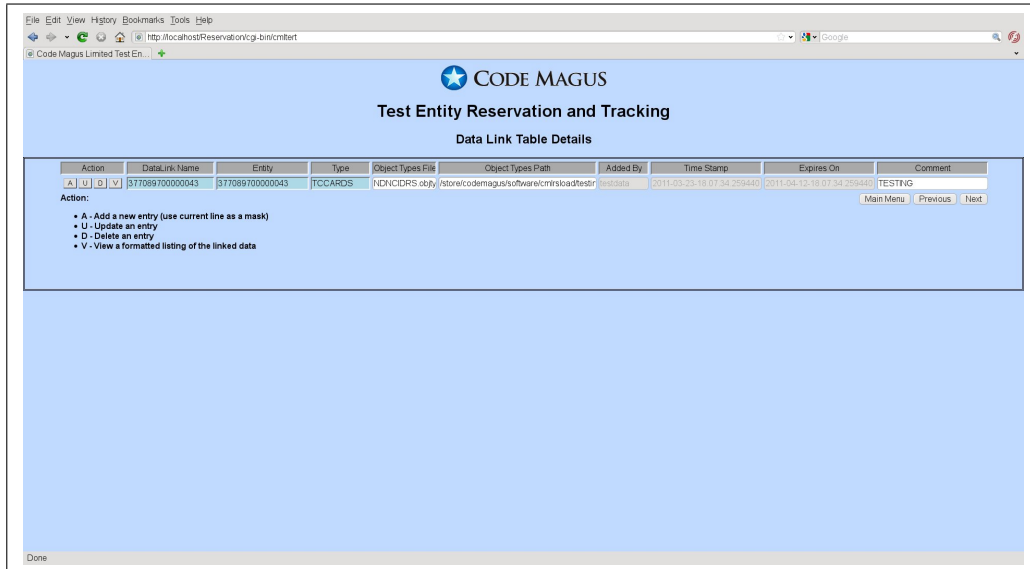


Figure 6: Swapping From a Reservation to View Link Data

The fields that may be updated are:

- Purpose. This is descriptive text that should describe the purpose or use of the entity.
- Group. The entity may be moved from one group to another. The new group must already exist.

### 3.6 Working with Reservation Linked Data

This page displays, up to a configured page maximum, all the reservation link data entries that have been made and that match the mask entered on the main menu. If there are more reservation link data items than will fit in the maximum allowed then the buttons 'previous' and 'next' can be used to navigate back and forward through the results.

Any user logged on to the system may create a reservation link data item, delete their own reservation link data and view all the reservation link data with out any special access authorisations. In order to update or delete a reservation link data item that does not belong to the logged on user then those specific authorisations need to be given on the access table maintenance page, see section 3.8 on page 18

The screenshot shows a web browser window displaying the 'CODE MAGUS' interface. The page title is 'Test Entity Reservation and Tracking' and the sub-header is 'Data Link Table Details'. Below this is a table with columns: Action, DataLink Name, Entity, Type, Object Types File, Object Types Path, Added By, Time Stamp, Expires On, and Comment. The table contains 20 rows of reservation data. At the bottom left, there is an 'Action:' legend with symbols for Add, Update, Delete, and View. At the bottom right, there are navigation buttons for 'Main Menu', 'Previous', and 'Next'.

Action	DataLink Name	Entity	Type	Object Types File	Object Types Path	Added By	Time Stamp	Expires On	Comment
[A] [U] [D] [V]	37708970000043	37708970000043	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:34.259440	2011-04-12-18:07:34.259440	TESTING
[A] [U] [D] [V]	37708970000050	37708970000050	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:33.179543	2011-04-12-18:07:33.179543	TESTING
[A] [U] [D] [V]	37708970000078	37708970000078	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:29.916450	2011-04-12-18:07:29.916450	TESTING
[A] [U] [D] [V]	37708970000084	37708970000084	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:33.177001	2011-04-12-18:07:33.177001	TESTING
[A] [U] [D] [V]	37708970000118	37708970000118	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:31.102091	2011-04-12-18:07:31.102091	TESTING
[A] [U] [D] [V]	37708970000209	37708970000209	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:30.591599	2011-04-12-18:07:30.591599	TESTING
[A] [U] [D] [V]	37708970000308	37708970000308	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:29.897893	2011-04-12-18:07:29.897893	TESTING
[A] [U] [D] [V]	37708970000340	37708970000340	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:32.194212	2011-04-12-18:07:32.194212	TESTING
[A] [U] [D] [V]	37708970000373	37708970000373	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:34.512229	2011-04-12-18:07:34.512229	TESTING
[A] [U] [D] [V]	37708970000423	37708970000423	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:29.910792	2011-04-12-18:07:29.910792	TESTING
[A] [U] [D] [V]	37708970000464	37708970000464	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:34.235671	2011-04-12-18:07:34.235671	TESTING
[A] [U] [D] [V]	37708970000821	37708970000821	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:31.026902	2011-04-12-18:07:31.026902	TESTING
[A] [U] [D] [V]	37708970000837	37708970000837	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:32.655354	2011-04-12-18:07:32.655354	TESTING
[A] [U] [D] [V]	37708970000936	37708970000936	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:32.375554	2011-04-12-18:07:32.375554	TESTING
[A] [U] [D] [V]	37708970001170	37708970001170	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:31.309017	2011-04-12-18:07:31.309017	TESTING
[A] [U] [D] [V]	37708970001983	37708970001983	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:31.026495	2011-04-12-18:07:31.026495	TESTING
[A] [U] [D] [V]	37708970002924	37708970002924	TCCARDS	NDNCIDRS.cdy	/store/codemagus/software/cmrisloadtestr	testdata	2011-03-23-18:07:31.946526	2011-04-12-18:07:31.946526	TESTING

Action:  
 • A - Add a new entry (use current line as a mask)  
 • U - Update an entry  
 • D - Delete an entry  
 • V - View a formatted listing of the linked data

Figure 7: Working with Reservation Linked Data

Reservation link data items are maintained on this page through selecting (by clicking) the applicable mini-button to the left of each row. As the cursor hovers over a button it will turn red. The action characters are:

- ‘D’ Delete. This will display a new page asking for confirmation to delete the selected reservation.
- ‘U’ Update. This will display a new page for updating the reservation with any values that have been changed. The key fields and automatically defaulted values can not be changed; rather a new reservation must be added. Values that are automatically defaulted are the user ID that is making the reservation and the time stamp.
- ‘A’ Add. This will display a new page for making a reservation using the values from the selected reservation as a template. One or more reservations can be added

one at a time using the add page and once all the required reservations have been added the user can return to the result page or the main menu.

- ‘V’ Show Linked data item. This option formats and displays the actual data record used to make the associated reservation. See figure 8 on page 14 for an example screen shot.

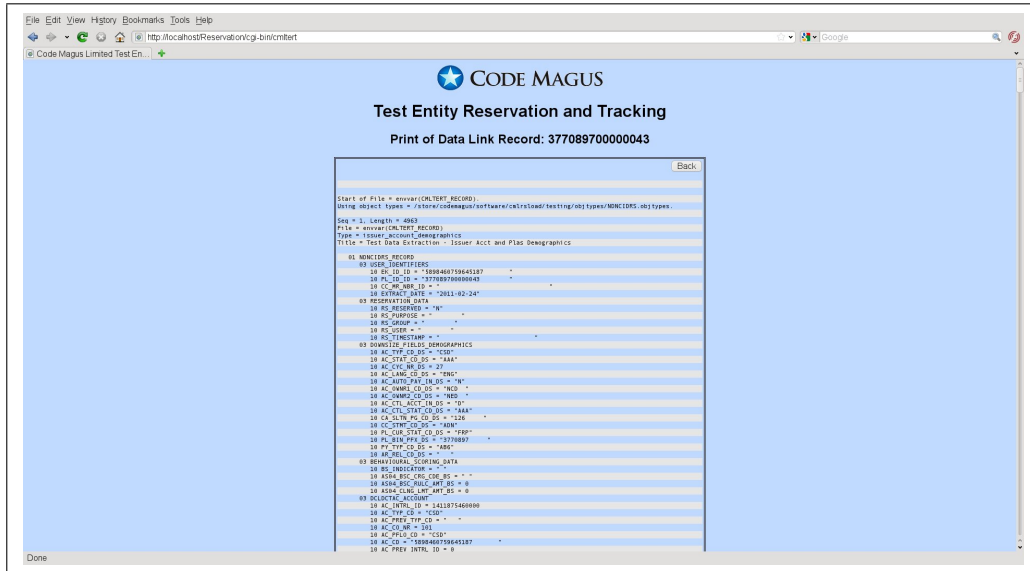


Figure 8: Working with Reservation Linked Data Detail

The fields that may be updated are:

- Object Types File. The Code Magus object types file definition name that is used when formatting the data record for display.
- Object Types Path. The path (or directory) in which the object types file resides. Together with the Object Types File name this makes the fully qualified file name of the object types definition file.
- Comment. Some descriptive text that describes and explains this entry.

### 3.7 Working with Reservation Pool Data

To look for test data that can be reserved and used from the pool of loaded data click on the 'Search for Test Data' button on the main menu. As shown in figure 9 on page 15 the search dialogue is then presented.

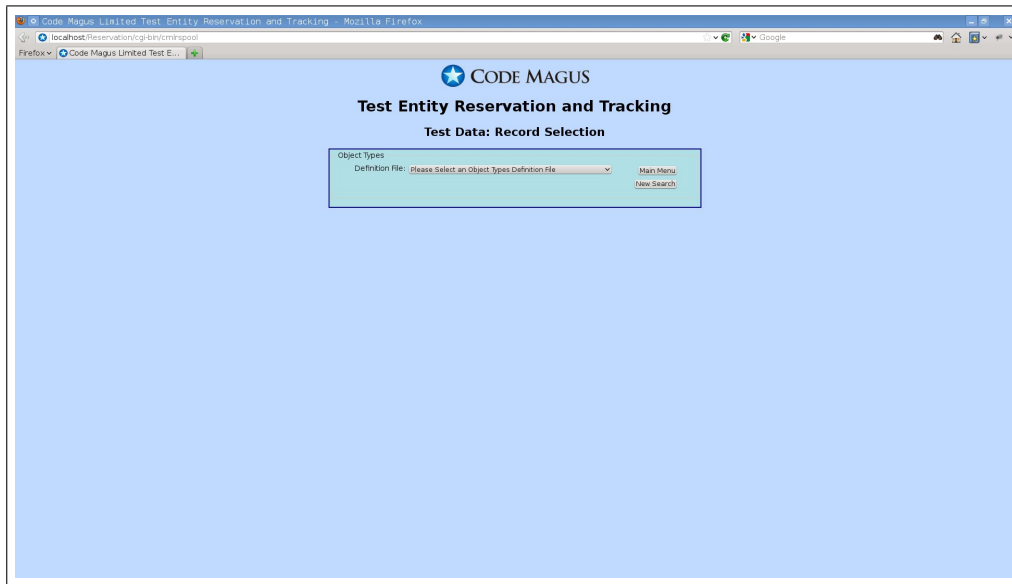


Figure 9: Searching for Test Data to Use

Select the object types definition that matches the test data you wish to work with and then the object type from within that. As each item is selected the dialogue is updated to the point as seen in figure 10 on page 16.

Now a conditional statement, that qualifies that result set of data returned from the search, may be entered. In the conditional statement all normal *expeval*[1] syntax and semantics apply, except for references to field names contained in the object types. For field names a dollar variable is supplied instead as a placeholder and must comprise a '\$' sign followed by the name and terminated with a white space. For example if the data being searched has credit card numbers in it and you want card numbers that start with 541281 then the statement would be written as:

```
$card like "^541281.*"
```

where the `"^541281.*"` is a regular expression (for the like operator) and means select all `$card` values that start with 541281 followed by any number of characters. If there is a requirement for the number to include only numeric digits the conditional statement could be written as:

```
$card like "^541281[0-9]+\+"
```

Once the conditional statement is complete click on 'resolve' which will add, to the dialogue, drop down selection lists where you can resolve all the dollar variables to



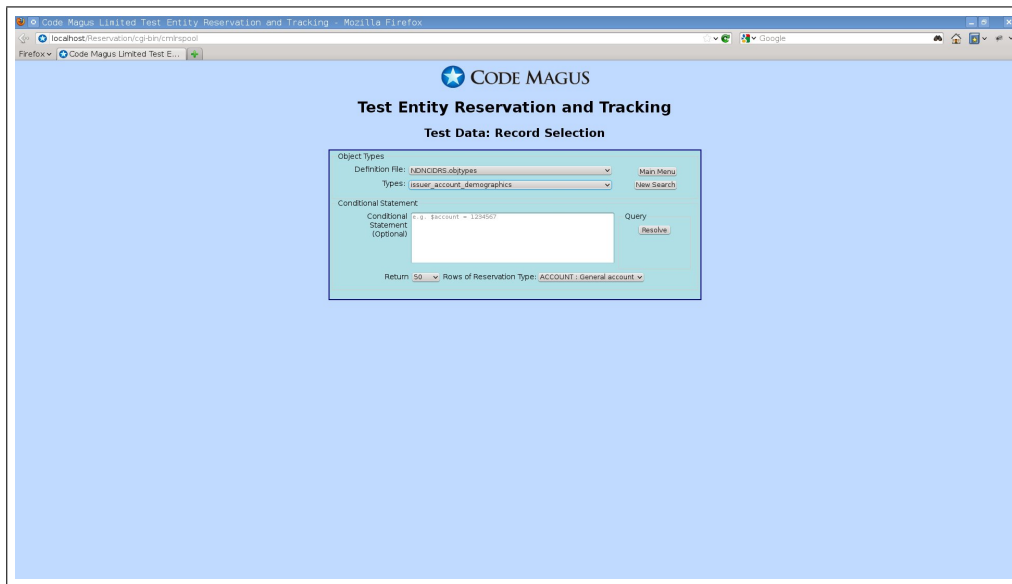


Figure 10: Specifying Search Criteria for Searching the Pool

actual field names within the object type as seen in figure 11 on page 16. The buttons ‘show’ and ‘submit’ are also added to the dialogue at this point.

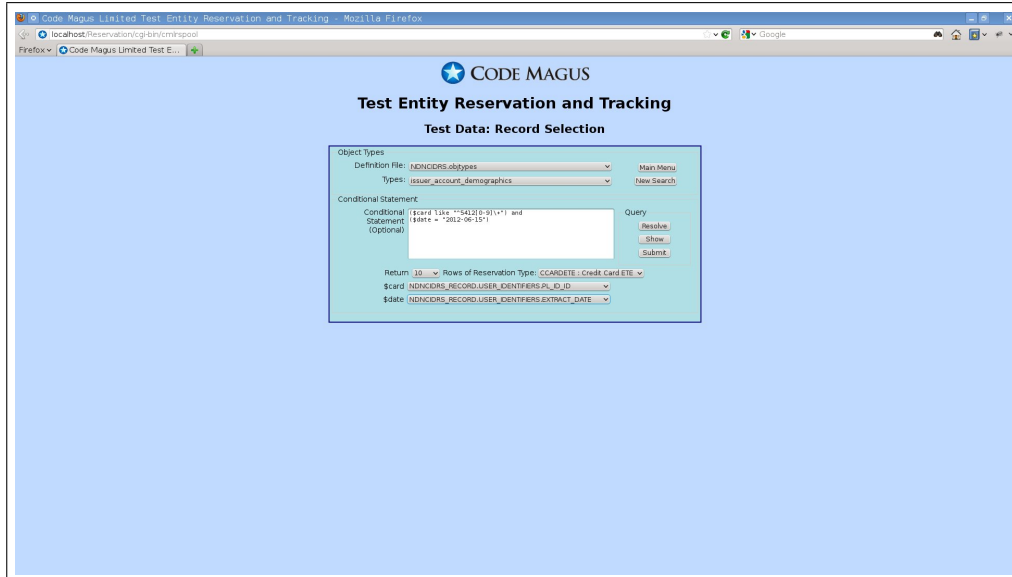


Figure 11: Specifying Criteria for Searching the Pool

In order to process the query correctly check the following points and once satisfied that they are correct click on ‘submit’.

- For each dollar variable select the fields that should resolve their name.
- Decide on the maximum number of results to be returned; If more are required

then the response will be correspondingly slower to return.

- Select the reservation type that the pool entries were added to the pool with from the select drop down list.
- Click on ‘submit’.

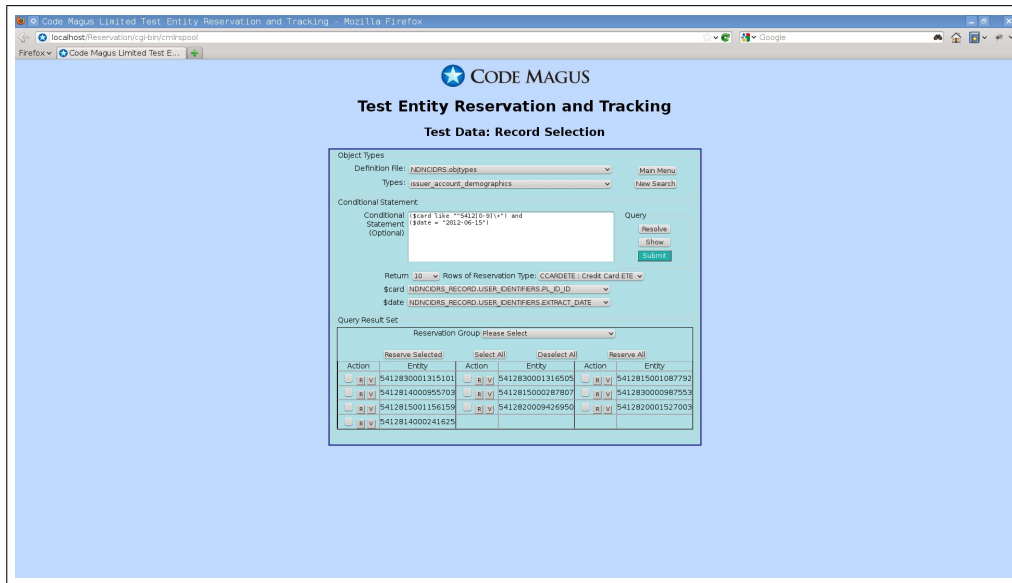


Figure 12: Working with Pool Data

The results of the query is a table of entities that are candidates for reserving. These candidates may already be reserved, but if they are not the dialogue allows the user to reserve them now. In order to reserve an entity, first select the ‘Reservation Group’ using the select drop down above the entities and then reserve entities in any of the following ways:

- To reserve one entity click on the mini button under ‘Action’ marked with an ‘R’.
- To reserve more than one entity, select each one by checking the radio button under ‘Action’ and then clicking ‘Reserve Selected’.
- To reserve all the entities in the result set click on ‘Reserve All’.

The buttons ‘Select All’ and ‘Deselect All’ allow the entire result set of entities to be checked or unchecked respectively.

The mini button marked with a ‘V’ under ‘Action’ allows you to view the entity data record that is held in the pool.

### 3.8 System Authorisation

Only users that have the update authorisation to the access table will see the 'Manage User Authorisations' option on the main menu and be able to navigate to this page. Access should be granted to the system administrator only.

This page displays, up to a configured page maximum, all the users and their authorisations that have been defined and that match the mask entered on the main menu. If there are more users than will fit in the maximum allowed then the buttons 'previous' and 'next' can be used to navigate back and forward through the results.

Action	User ID	Type	Group	Access	Primary	Linkdata	Modified By	Time Stamp	Password	Comment
A U D	payward	F	F	N	N	N	rsadmin	2011-02-25-10-11-07-101174	*****	Patrick H
A U D	patrick	F	F	N	N	N	rsadmin	2011-02-25-10-12-30-689564	*****	Patrick P
A U D	rsadmin	N	N	F	N	N	DB282	2011-02-25-10-09-59-984090	*****	Reservation System User Administrator
A U D	stephen	F	F	N	N	N	rsadmin	2011-02-25-10-12-15-324080	*****	Stephen D
A U D	testdata	F	F	N	N	N	rsadmin	2011-02-25-10-15-18-120590	*****	For Batch Testing

Action:  
 • A - Add a new entry (use current line as a mask)  
 • U - Update an entry  
 • D - Delete an entry

Figure 13: Maintaining Authorisations

User IDs are maintained on this page through selecting (by clicking) the applicable mini-button to the left of each row. As the cursor hovers over a button it will turn red. The action characters are:

- 'D' Delete. This will display a new page asking for confirmation to delete the selected user ID.
- 'U' Update. This will display a new page for updating the user ID with any values that have been changed. The key fields and automatically defaulted values can not be changed; rather a new user must be added. Values that are automatically defaulted are the user ID that is modifying the user ID entry and the time stamp.
- 'A' Add. This will display a new page for adding a user ID using the values from the selected user ID as a template. One or more user IDs can be added one at a time using the add page and once all the required user IDs have been added the user can return to the result page or the main menu.

The fields that may be updated are:

- Access. This defines the authorisation that the user has in maintaining each of the four tables in the system. Each table may be set individually, they are:
  - Primary. The reservations table in which all entities are recorded when reserved. The authorisations given here are usually for administrators that may need to correct entries that are incorrect or out dated. There is no need to grant any authorisations on this table to normal users of the system as any user may make a reservation (that is not already made), view all other reservations and delete their own reservations.
  - Type. The types table defining the types that constrain the entities. Access to update or delete types should only be given to an administrator of the system.
  - Group. The groups table defining the groups in which entities may be placed. Access to update or delete groups should only be given to an administrator of the system.
  - Access The user authorisation table (this one).

The user has separate authorisation settings for each table and the values are as follows:

- ‘F’ - Full. This setting allows the user to delete, update and read rows from this table.
  - ‘U’ - Update. This setting allows a user to update and read rows from the table.
  - ‘R’ - Read. This setting allows users to read rows from the table.
  - ‘N’ - None. This setting allows the user no access to any operation on the table.
- Comment. This is descriptive text that should describe the user and their authorisation settings.

## 4 Installation

The following sections detail the platforms and web servers that the Testament system runs under and how to install it on them.

### 4.1 Apache on Linux using DB2

To install the Testament system on Linux running under Apache and using DB2 for Linux perform the following steps:

#### 4.1.1 Set the Code Magus Home Directory

All Code Magus software must be installed in one place. On Linux this is usually `/usr/local/CodeMagus`. The environment variable `CODEMAGUS_HOME` must contain the name of this directory.

#### 4.1.2 Unpack the Installation Package

Obtain the compressed installation file and unpack it in the directory `CODEMAGUS_HOME` as follows:

```
[root@codemagus ~]# cd ${CODEMAGUS_HOME}
[root@codemagus CodeMagus]#tar zxvf \
    CODEMAGUS_RHEL50_____I686_____PRODUCT_BIN_CMLRESERVE_D20101005_T1809.tgz
```

#### 4.1.3 Copy Apache Configuration File

Copy the file `Reservation/etc/CodeMagus_Reservation.conf` to the Apache configuration directory. This is usually `/etc/apache2/conf.d/`; consult the Apache documentation and the machine and Apache configuration to determine this directory if it is not.

#### 4.1.4 Create the DB2 Tables

Identify the DB2 instance under which the system will be installed and create the tables required. Templates of data definition language (DDL) statements for creating the schema, table space and each table are supplied in the directory `CODEMAGUS_HOME/Reservation/ddl/`. Each file needs to be checked for any customisation that may be required, following which the statements can be executed with the commands:

```
db2 connect to sample user DB2_admin_user using DB2_admin_password
db2 -vtf <DDL_filename>
db2 connect reset
```

where `DB2_admin_user` and `DB2_admin_password` are the user ID and password respectively of a user that has the authority in the chosen DB2 instance to create schemata, table spaces, tables and grant authorities to users. The connect and disconnect statements could be copied into the top and bottom of each DDL file in which case only the one command is required or use 'su' to switch to the DB2 administrator user and run the scripts directly from there.

The template files are as follows and should be run in the order given:

- `create_schema.sql` Creates the DB2 schema for the Testament system. If you change the name of the schema or plan to use an existing one then all the following DDL statements must be changed to reflect this.
- `create_tablespace.sql` Creates the table space for the system. The path to the actual data file needs to be set according to site standards and naming. All other parameters may also be changed, but if the name of the table space is changed then this must be reflected in the table create DDL statements. This is an optional step if an already existing tablespace will be used. The existing tablespace must have the correct page size.
- `create_tablespace_temp.sql` Creates a temporary table space for the reservation system. A temporary table space with a large `PAGESIZE` must exist for the reservation system to handle bulk deletes of the link data table which includes a Binary Large Object (BLOB). This is an optional step and is only required if the database does not already have a temporary tablespace of this configuration.
- `create_access.sql` Creates the access table. This table holds the access authorisations for each user.
- `create_type.sql` Creates the type table. This holds the types that constrain the entity names of a reservation.
- `create_group.sql` Creates the group table. This holds the group names in which entities are categorised.
- `create_primary.sql` Creates the primary table. This table holds the actual reservations made.
- `create_linkdata.sql` Creates the linkdata table. This table holds the linked data information for each reservation that has had linked data loaded for it.

## 5 Customisation

The Testament system requires some customisation and is described for each platform below.

### 5.1 Apache on Linux using DB2

To customise Testament on Linux running under Apache perform the following steps:

#### 5.1.1 Configure the Apache Settings

Edit the configuration file that was copied during installation `/etc/apache2/conf.d/CodeMagus.Reservation.conf` as follows:

- Code Magus Base Directory

Edit all occurrences of the string `RESERVATION_INSTALL_BASE` to the value found in `CODEMAGUS_HOME` (usually `usr/local/CodeMagus` as set previously).

- DB2 Install Base Directory

Edit all occurrences of the string `DB2_INSTALL_BASE` to the directory name where DB2 is installed.

- DB2 Instance name

Edit the string `DB2_Instance_name` to the name of the DB2 instance that the tables have been created in.

- DB2 Database name

Edit the string `DB2_Database_name` to the name of the DB2 database that the tables have been created in.

- DB2 Schema name

If the chosen schema is not `CMLRT00` then it can be changed here to the same name as was used when creating the tables for the system.

- DB2 User name

Edit the string `DB2_Connection_user_name` to the user of the DB2 user that will be used to connect to DB2 with.

Note that there are restrictions that DB2 imposes on the length of a user name. Currently this is 8 characters, but consult your DB2 documentation for confirmation.

- DB2 Password

Edit the string `DB2_Connection_user_password` to the password of the DB2 user that will be used to connect to DB2.

Note that there are restrictions that DB2 imposes on the length of a password. Currently this is 14 characters, but consult your DB2 documentation for confirmation.

- DB2 Default values

As a convenience (not usually for a live system) the DB2 user ID may be automatically filled in on the log in page by setting the following environment variable to the values required.

`DB_DEFAULT_UID` for the database user ID.

- Reservation system default values

The number of results per page can be set so that the user does not have to scroll the page down; i.e. results are limited to a page of data only and the user can use the 'previous' and 'next' buttons to retrieve more data. Set the following environment variable to a valid integer.

`CODEMAGUS_RT_LINES_PER_PAGE`

- Apache Directory Settings

Check and Confirm that the attributes set for the CGI binary and HTML directories are correct. These are set up as alias directories under Apache so that web pages and executable programs do not have to be copied to other directories.

Apache needs to be restarted after this using the following command from `root`:

```
apachectl restart
```

...OR...

```
apache2ctl restart
```

### 5.1.2 Bind the DB2 Plan

The supplied DB2 package must be bound to the DB2 instance under which the system will be run. Note that the user running the bind and the owner of the tables and package (default `CMLRT00`) requires the correct authorisation in DB2 to be able to bind the package; see the next section [5.1.3](#) on page 24 for more information.

The DDL file `bind.sql` can be executed under DB2 as follows:

```
db2 connect to sample user DB2_admin_user using DB2_admin_password
db2 -tf Reservation/ddl/bind.sql
db2 connect reset
```



where `DB2_admin_user` and `DB2_admin_password` are the user ID and password respectively of a user that has the authority in the chosen DB2 instance to perform binds. The schema set in the `collection` keyword may be changed for site specific standards.

### 5.1.3 Grant DB2 Permissions

`grant.sql` is a template for granting DB2 level authorisation to the DB2 user used to connect to DB2 on behalf of the Reservation system users and for the owner of the system in order to execute the package and access the tables in the system. After modifying it to suit your installation standards the statements can be executed as follows:

```
db2 connect to sample user DB2_admin_user using DB2_admin_password
db2 -vtf Reservation/ddl/grant.sql
db2 connect reset
```

where `DB2_admin_user` and `DB2_admin_password` are the user ID and password respectively of a user that has the authority in the chosen DB2 instance to grant these authorities.

### 5.1.4 Add a Reservation System Administrator

`insert_access.sql` is a template for adding the first user to the reservation system. This should be user with full rights to update authorisation settings for all users on the Reservation system. The user may be changed, but do not change the password yet. When first logging on to the system with this user use the password `rsadmin`, but then change it immediately for security purposes. This user may then create other users and reset passwords via the on-line system.

```
db2 connect to sample user DB2_admin_user using DB2_admin_password
db2 -vtf Reservation/ddl/insert_access.sql
db2 connect reset
```

where `DB2_admin_user` and `DB2_admin_password` are the user ID and password respectively of a user that has the authority in the chosen DB2 instance to insert into the access table.

## A Apache Configuration File Template

```
# File: CodeMagus_Reservation.conf
# This Apache configuration file defines all the Apache settings required for
# the correct running of the Code Magus Ltd. Reservation system. It should be
# edited to reflect correct values for the options and then copied to the Apache
# configuration file include directory: This is usually /etc/apache2/conf.d/ .
# If it is not then consult the Apache documentation and the set up of the
# machine to determine where this is.
#
# $Author: hayward $
# $Date: 2011/01/28 13:36:44 $
# $Id: CodeMagus_Reservation.conf,v 1.6 2011/01/28 13:36:44 hayward Exp $
# $Name: $
# $Revision: 1.6 $
# $State: Exp $
#
# $Log: CodeMagus_Reservation.conf,v $
# Revision 1.6 2011/01/28 13:36:44 hayward
# Add definitions to support a single
# user connection to the database.
#
# Revision 1.5 2010/12/20 12:51:28 hayward
# Changes or additions to support Linkdata table.
#
# Revision 1.4 2010/10/07 21:25:29 hayward
# Added more comments
#
# Revision 1.3 2010/09/29 11:12:44 hayward
# Further changes from review 2:
# . Remove copyright sign
# . Improve english - use "add" instead of "insert"
# . Change scroll down/up to previous and next.
# . Add imagery.
# . Improve message on successful add.
# . Improve directions about usage of the action.
# . Confirm that cascade delete works for type and group.
#
# Revision 1.2 2010/09/27 21:14:32 hayward
# Major fixes from review:
# 1. Put in copyright.
# 2. Use a separate insert page for inserting a row in a table.
# 3. Use the type constraint for checking an entity.
# 4. Allow exact match lookups.
# 5. Limit the number of rows on a screen
# and implement scrolling.
#
# Revision 1.1.1.1 2010/09/20 11:05:38 hayward
# Initial CVS take on.
#
#
# 1. Edit all occurrences of *INSTALL_BASE to the correct path values.
```

## A APACHE CONFIGURATION FILE TEMPLATE

---

```
# 2. Edit PATH_TO_BASE_OF_FORMATS to a valid directort where object type files
# are stored. These are references to meta-data specified when the batch
# reservations and linked data is loaded and allows an online user to view
# the linked data record that was used to reserve a linked reservation.
# 3. Edit DB Instance name to the correct value for the DB2 instance the tables
# reside on
# 4. Edit the DB2 connection information. The reservation system always connects
# to DB2 using the same userid irrespective of the reservation system user
# logged on. This DB2 user must have read, update and delete access to all
# the tables of the reservation system. Actual authorisation to read, update
# and delete information from these table is maintained and controlled in
# the access table in the reservation system.
#
#####
# Global Code Magus Environment variables for diagnosis
#####
# Use these only under direction of Code Magus support
#SetEnv CODEMAGUS_MSGLEVEL VERBOSE
#SetEnv CODEMAGUS_OUTPUT_PREFIX 1
#SetEnv CODEMAGUS_OUTPUT_PREFIX_TS 1
#SetEnv CODEMAGUS_OUTPUT_SPEC text(/var/log/apache/reservation_log.txt,mode=w)
#
#####
# Code Magus Reservation environment variables for CGI
#####
# Code Magus Base Directory
SetEnv CODEMAGUS_HOME RESERVATION_INSTALL_BASE
# Code Magus Meta Data Base Directory
SetEnv CODEMAGUS_FORMATS PATH_TO_BASE_OF_FORMATS

# Reservation system HTML path
SetEnv CODEMAGUS_RT_HTML_PATH RESERVATION_INSTALL_BASE/Reservation/html
# Number of results per page to display
SetEnv CODEMAGUS_RT_LINES_PER_PAGE 20
#
# Code Magus Reservation environment variables for DB2 Path and instance
SetEnv LD_LIBRARY_PATH DB2_INSTALL_BASE/sql/lib/lib
SetEnv DB2INSTANCE DB2_Instance_name
#
#Set the DB name,user,password and schema name.
SetEnv CODEMAGUS_RT_SCHEMA DB2_Schema_name
SetEnv CODEMAGUS_RT_DBUID DB2_Connection_user_name
SetEnv CODEMAGUS_RT_DBPWD DB2_Connection_user_password
SetEnv CODEMAGUS_RT_DBDB DB2_Database_name
#
# The default user ID is used to prefill the login page with the user will still
# have to type in the password or over type the values with others they prefer.
# This is only useful if one user id is used for all users of the system
#SetEnv DB_DEFAULT_UID dft_User
#
#####
# Following are the Apache configurations for accessing the CGI programs
```

```
# and HTML pages.
#####
# Code Magus Limited Reservation HTML and CGI directories as aliases
Alias /Reservation/html/     RESERVATION_INSTALL_BASE/Reservation/html/
Alias /Reservation/cgi-bin/  RESERVATION_INSTALL_BASE/Reservation/cgi-bin/
#
# Properties for HTML files.
<Directory RESERVATION_INSTALL_BASE/Reservation/html>
    AllowOverride All
    Options +FollowSymLinks +Includes +Indexes
    Order allow,deny
    Allow from all
</Directory>
#
# Properties for CGI scripts.
<Directory RESERVATION_INSTALL_BASE/Reservation/cgi-bin>
    SetHandler cgi-script
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
#
```

## References

- [1] expeval: Expression Evaluation API Reference. CML Document CML00052-01, Code Magus Limited, November 2009. [PDF](#).
- [2] Testament: Batch Utilities Version 1. CML Document CML00072-01, Code Magus Limited, October 2010. [PDF](#).