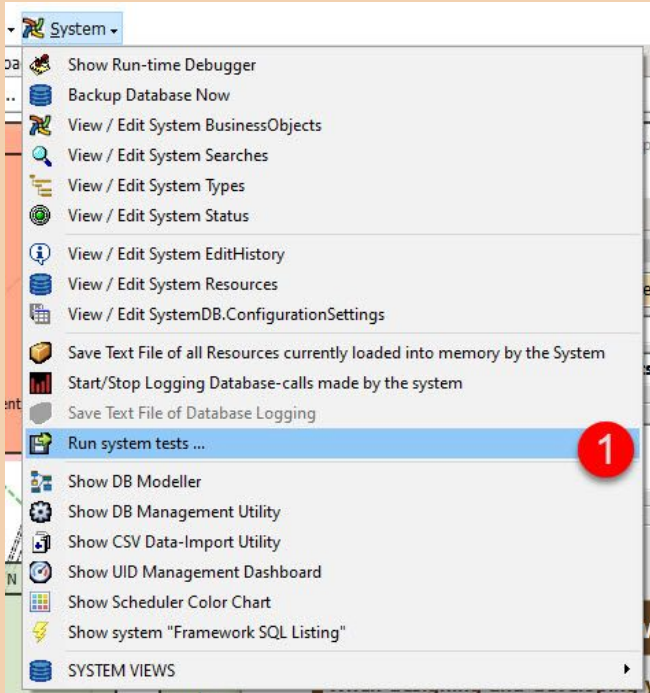# The System "Run Tests" window

When designing and developing your App you will add many SQL scripts to show user-interface elements such as grids, reports and dashboards.

As a system grows, changing one element may impact on another, leading to unexpected bugs. Orixa includes a mechanism to methodically test all or parts of the system-SQL in your app to make sure it works.
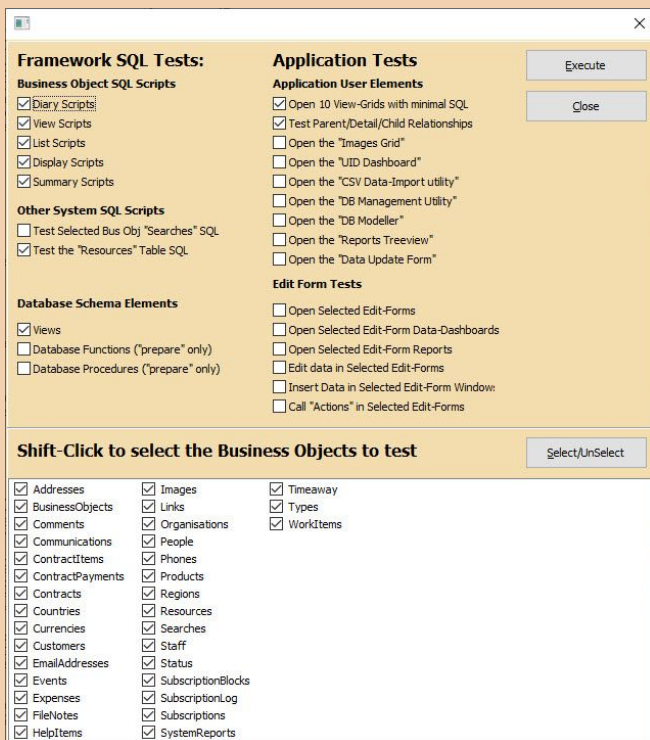
Accessing and viewing the "run tests" window

Run system tests

To access the "run tests" window, select "Run System Tests" (1) from the System Menu.

Note that it is also possible to access this window from the System DB Modeller.

Run system tests window

Run System Tests

This window allows Admin users to select automated tests and run them.

Running most tests simply checks the operation of SQL statements in the System-Tables.

A few of the tests will actually open user-interface elements such as Grids etc., to allow the Admin user to confirm these are all working properly,

Elements of the Run System Tests Screen

Run system tests window Annotated

1. Select which of the SQL Scripts in the BusinessObjects system-table you want to test.

2. Test all "Searches" (for the selected BusinessObjects) and test all SQL in the "Resources" system-table.

3. Test meta-data tables such as Views, Functions and Procedures. Note that Functions and Procedures are only "Prepared", since executing them might result in complex consequences for your App.

4. User interface tests: Select these to check that expected elements of the user-interface are working properly.

5. Edit Form tests. Running these tests allows you to open multiple windows and automate the process of adding and editing data to test for any erorrs.

6. Click to actually start tests running.

Why test, and what can tests show?

When you rename fields in your App this may have consequences if the field-name is referenced by a Resource, BusinessObject, View, Function, Procedure or other App SQL. Tests will immediately show you parts of your App which have been damaged by such changes.

Testing Edit Forms can be useful to check that constraints, generation and computation options all work well. Errors in the SQL of these parts of your App will immediately be shown by the testing system.

Actually running tests
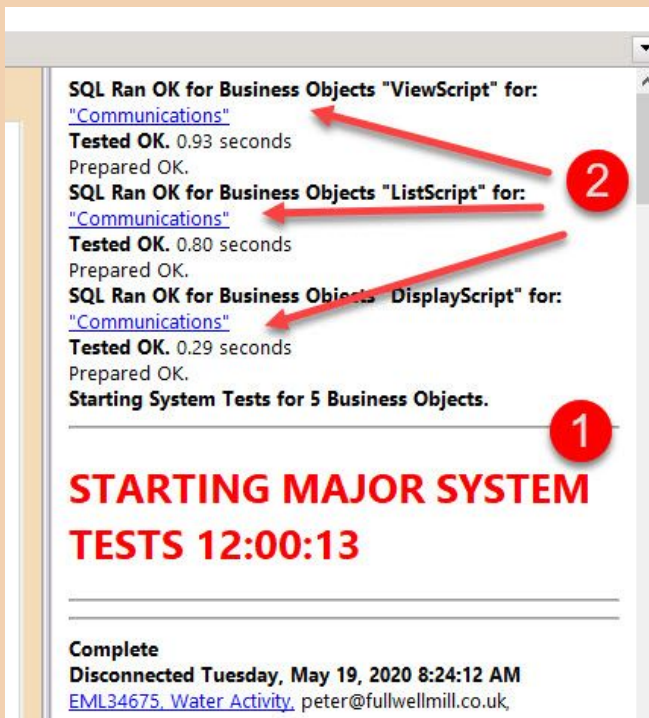
Run system tests message Log Start

Once you have selected which tests to run and clicked "Execute" your App will start to run the SQL and open user-interface for the different App elements.

A log of everything that has been done is shown in the Application Message Log (1).

Note that as items are tested you are given useful information about each script (2). For example the time a script took to run. If this is long a warning will be added to the message log, and you can review the script.

Note that the blue underlined links will take you directly to the specific system-table-record being tested.
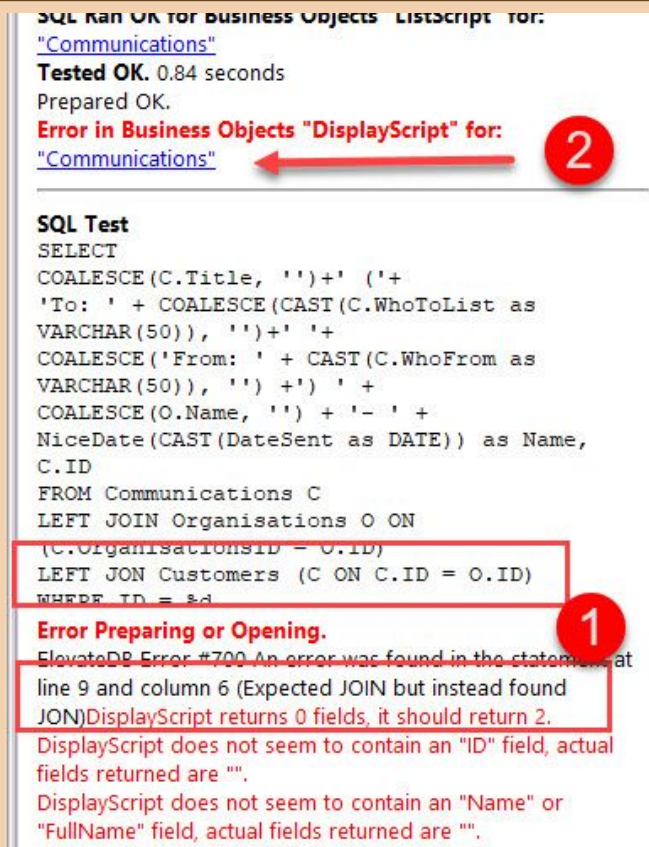
Run system tests message Log with Error

Example of an Error Message

If your App includes badly written SQL the testing system will show you this, and try to provide useful error messages about it.

The image shows a script in which the JOIN statement has been badly written. (1) The author has written JON instead of JOIN.

Note how the error message expresses: "Expected JOIN but instead found JON" and gives the line-number for the error.

To fix this problem, click on the link (2) and the edit-window for this BusinessObject will open. Find the "DisplayScript" and change "JON" to "JOIN" ...

Tests Complete

The message log will hold a list of **all** tests. You can scroll down through it to see the full details of the testing, and save the output to a file on-disk if you wish.

Note that the final step includes running a set of minor tests to ensure that your App is well-formed. These are shown at (1).

MAJOR SYSTEM TESTS COMPLETED 12:01:55

Not all BusinessObject records have "Searches" records, please add searches for the followingtables:

All "Searches" records have LinkTables which are "Normal", "Master", "Detail", "Child" or "Hidden" BusinessObjects.

All "Status" records have LinkTables which are "Normal", "Master", "Detail", "Child" or "Hidden" BusinessObjects.

All "Types" records have LinkTables which are "Normal", "Master", "Detail", "Child" or "Hidden" BusinessObjects.

SystemDB.ConfigurationSettings table is present in BusinessObjects.

SQL Ran OK for ViewScript:
"SystemDB.ConfigurationSettings"
Tested OK. 0.49 seconds
Prepared OK.
All "parent" constraints" have correctly named "child" BusinessObjects .

The tables: "Subscriptions" are NOT marked in the BusinessObjects table as "BusObjType = C" , but have "Parent" constraints.

The tables: "SystemReports" are in the BusinessObjects table, but not present in the database.

The tables: "GoogleDetails, GoogleCalendar" are not included in the BusinessObjects table.

Run system tests message Log Complete