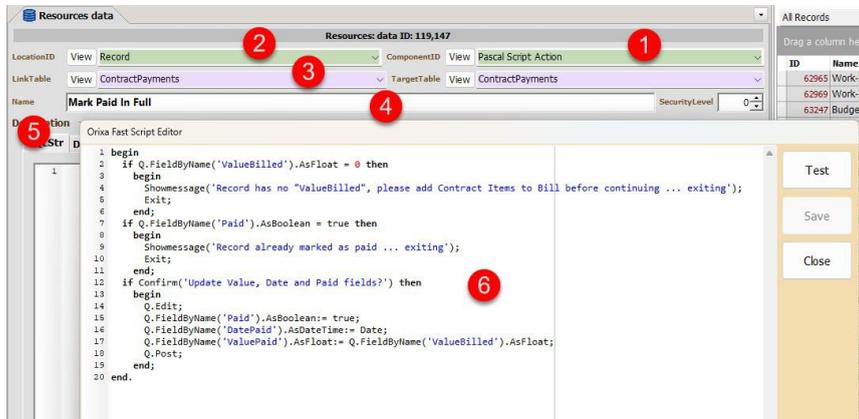


How to add Pascal Scripts in your App

A more detailed version of this item is available here:

[Dynamic Pascal-Scripting in Orix: Adding a Resource to create Rapid-Entry-Grids in your App](#)

Pascal Scripts are added to "ObjectProperties" of Resources



Steps to create Rapid Entry Grid

1. Create a Resource with ComponentID = "Rapid Entry Grid" or "Pascal Script Action"
2. Set its location to the location where you want the Action to appear in your App: "Entity", "Record", "System" etc.
3. Select a "LinkTable" and "TargetTable", this defines exactly where the Action will appear (the "LinkTable"), and passes the ID and Table-Name (of the "Target Table") to the resource when it is called.
4. Give a name to the Resource, this name will appear as an item in the menu displayed for the user to click.
5. If you are creating a Rapid Entry Grid resource, type SQL to generate a data-grid the user can view and edit. (Do not add SQL for a "Pascal Script Action").
6. From the Actions Menu, open the "ObjectProperties" window for the Resource. Write a **Pascal Script**.
7. For a Rapid Entry Grid, this script will run when the user clicks the "Post" button on the Rapid Entry Grid.
8. For a Pascal Script Action the script you have written will run as soon as the user clicks the item from the Actions Menu.

Explaining the Script

```
begin
  if Q.FieldByName('ValueBilled').AsFloat = 0 then
    begin
      Showmessage('Record has no "ValueBilled", please add Contract Items to Bill before continuing
... exiting');
      Exit;
    end;
  if Q.FieldByName('Paid').AsBoolean = true then
    begin
      Showmessage('Record already marked as paid ... exiting');
      Exit;
    end;
  if Confirm('Update Value, Date and Paid fields?') then
    begin
      Q.Edit;
      Q.FieldByName('Paid').AsBoolean:= true;
      Q.FieldByName('DatePaid').AsDateTime:= Date;
      Q.FieldByName('ValuePaid').AsFloat:= Q.FieldByName('ValueBilled').AsFloat;
      Q.Post;
    end;
```

end

- All scripts are enclosed with "begin" and "end".
- The script checks the values of several fields in the record using the syntax "Q.FieldName('something').AsFloat". Depending on the values of these fields it may exit without taking any action. These lines are **protective programming**.
- The Function "Confirm('A question string')" is used to allow the user to confirm they really want the subsequent lines of code to run.
- The values in the record are Updated using the syntax:
Q.FieldName('FieldName').AsFloat:= Some value.
The local variable "Date;" is used to reference the current date.

An Example "Rapid Entry Grid" Resource Script

The following Pascal is a working example of a simple script, copied from the example in the image above. In an Orixia App the script is stored in the "ObjectProperties" of your Resources data-table. Because the Resource is a Rapid Entry Grid the script will be passed references to the data in the grid. This data is referred to by the "Q" object in the script. "Q" here stands for "Query." The programmer can write code to reference the data in the Grid directly, and can update and change data programatically from the script.

```
var
  i : integer;
begin
  Q.First;
  for i := 0 to DM.RecordCount - 1 do
    begin
      if Q.FieldName('Complete').AsBoolean = true then
        RunSQL(FORMAT('UPDATE WorkItems SET Complete = true ' +
          ' WHERE ID = %d ', [Q.FieldName('ID').AsInteger]));
      Q.Next;
    end;
  end.
```

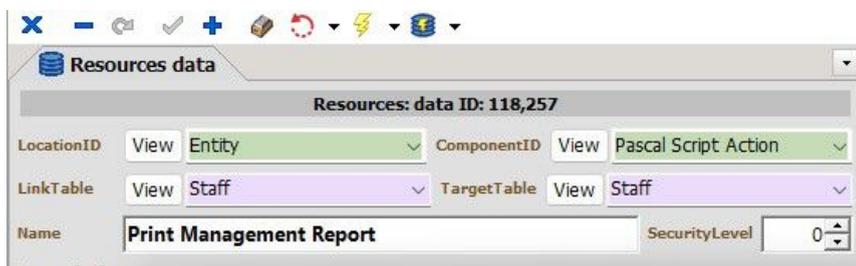
What does the Script do?

It jumps to the first item in the Grid (Q.First), then checks if the user has ticked the "Complete" box. If so it runs a SQL statement which updates the data in the Orixia Database for your App. Then it calls "Next" to move to the next record and repeat the process. The "for loop" ensures that the code runs for every row in the grid, to ensure that all the records are checked and updated as necessary.

There is more information on how to write Pascal Scripts here:

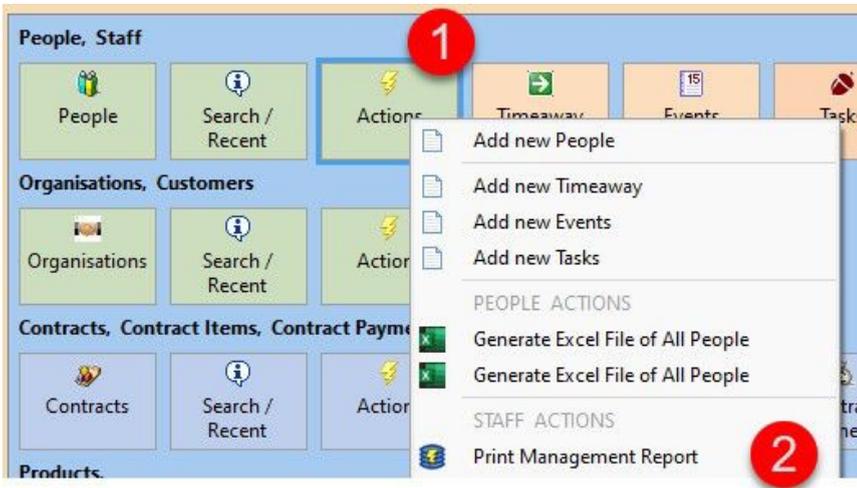
[Writing Pascal Scripts](#)

An Example of a working Pascal Script Action Resource



Pascal Script Action Resource Example

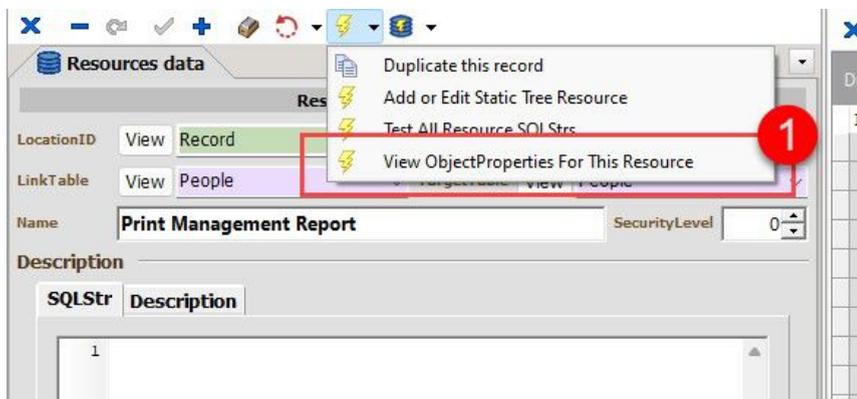
The above image shows a Resource Record added with settings needed to create a new "Action" in your Orixia App.



How the Pascal Script Action appears in the App

The above image shows how the the Action will appear in your app (at 2.) in the Actions menu (shown at 1.) of the "People, Staff" Entity. Note that because the "LinkTable" is set to "Staff" the Action appears under the heading "STAFF ACTIONS"

Object Properties of the Pascal Script Action Resource



Access the "ObjectProperties" of the Resource

The Pascal Script which will run when the user triggers the Action is stored in the "ObjectProperties" of the Resource. Open a window to edit this data by clicking "View ObjectProperties for this Resource" as shown above.



Pascal Script Example

1. Resource is set with "ComponentID" of "Pascal Script Action".
2. The Name of the Resource is set to something which the user can understand as a cue for what happens when the Resource runs.
3. A Pascal Script is added in the "Object Properties" of the Resource and can be viewed, edited and tested.

Explaining the example script

```

var
aFileName: String;
begin
  aFileName:= InputBox('Please add a filename', 'Filename:', '');
  Showmessage(aFileName + ' will be generated please wait ... ');
  if aFileName <> '' then
    RunScript(' CALL PDFRunner(''OrixaBilling'', 236, '' + aFileName + ''); ')
  else
    Showmessage('No file-name provided.');
```

end.

The above script asks the user for to provide a filename and then uses the key-word "RunScript" to call a database procedure "PDFRunner" and creates a PDF file saving it with a file-name chosen by the user. Note that for such a script to work in your App your database would need to include the "PDFRunner" functionality.

Adding PDF Runner functionality is explained here: [Adding "PDFRunner" and "EmailRunner" Dynamic Link Libraries to automate creation of PDFs and emailing files from your Orixa App](#)

The script declares a variable "aFileName" to hold a value entered by the user. The method "InputBox" asks the user to enter a value for the "aFileName" variable. If the user has entered a file-name, then the method "RunScript" will be called.

The "RunScript" method is a powerful tool for Orixa Actions. It allows **any** procedure written for your database to be called from an Action. All that is needed is to draft a "RunScript" call correctly. The process of writing these calls is technical, but once understood is fairly straight forward.

Note that an action can contain multiple sections with multiple "RunScript" lines. Also note that a pascal script can contain checks and verifications such as asking the user to say "yes", pick options or enter data.

Calling a database procedure from a Pascal Action Script

1. Enclose the "call" in the RunScript method: RunScript(' <add call here> ');
2. The call should be in the form: CALL ProcedureName(<list of parameters>);
Note that the final semi-colon is essential
3. The list of parameters must be well formed. If it is not the call will fail with an error message.
4. Entering parameters within a Pascal Script:
 - Each parameter must be separated by a comma.
 - Numbers can be entered simply.
 - String data (Varchars) must be enclosed in two single-quotes. As shown with "OrixaBilling" in the above example. This is called "escaping" the single-quote, and is needed because the call is itself string data, starting and ending with a single-quote, so the compiler needs to see 2 single-quotes in a row to know it has not come to the end of the string-data.
 - When a variable is passed into the middle of the string (as with the aFilename variable in the above example), the single-quotes must be increased to 3. This is because you are concatenating multiple strings and adding in variables from the outer part of the script.