Dynamic Scripting in Orixa: Adding a Resource to create Rapid-Entry-Grids in your App

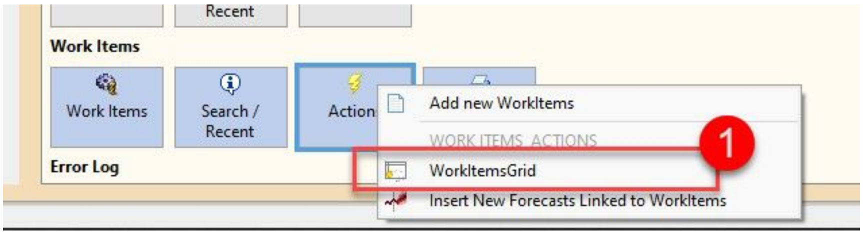
> **Orixa has built in systems for data entry and data management which are reasonably easy and intuative. However where large quantities of data are being managed, it is frequently much more economical to write systems for rapid-data entry. These save staff a lot of time, and save management a lot of money, as they allow hundreds or thousands of data-rows to be posted into the system database in an automated or semi-automated manner.**
> **If you have your own Custom Orixa App (built by Orixa engineers) these types of features can be added to your App to meet any requirement. It is also possible to add rapid entry systems yourself, by adding Resources to your App. This is a technically complex process, but can be done by any User of an Orixa system if they have a high enough Security Level and understand the required steps.**

When to use a Rapid Entry Grid Resource

When many rows of data are being processed by users with relatively minimal interaction: For example reviewing data imported from another Application such as Excel, prior to posting it into your App.

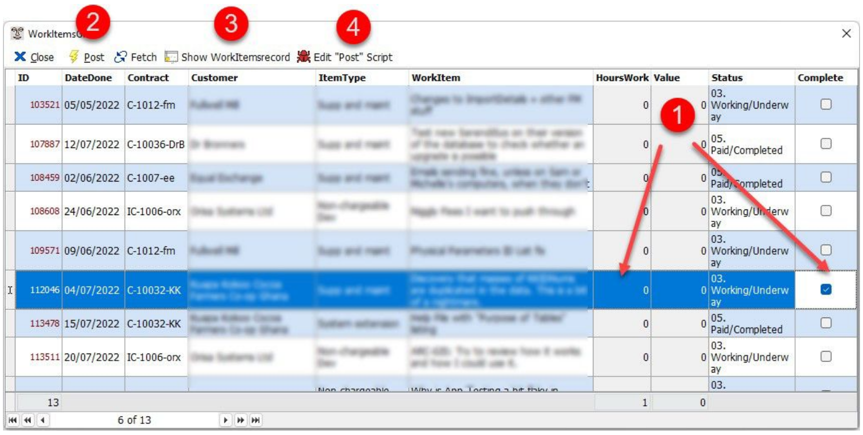
When you wish to post updates into multiple BusinessObjects simultaneously from a single user action. Most data-entry in Orixa occurs at the "Record" level (for example "Farmer" or "Customer"). If have data for a Farmer, coupled with data for their agricultural production, or data for a Customer combined with data of their orders it may be very time-consuming for staff to manually key all this data into your system. A rapid entry grid can automate this process, and add it to your App with a few key-presses.

An Example of a Rapid Entry Grid Resource



Rapid Entry Grid Resource

> 1. Rapid Entry Grids appear in the "System", "Entity" or individual record "Edit-Form" Action-menu's .



ID	DateDone	Contract	Customer	ItemType	WorkItem	HoursWork	Value	Status	Complete
103521	05/05/2022	C-1012-fm	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☐
107887	12/07/2022	C-10036-DrB	[redacted]	[redacted]	[redacted]	0	0	05. Paid/Completed	☐
108459	02/06/2022	C-1007-ee	[redacted]	[redacted]	[redacted]	0	0	05. Paid/Completed	☐
108608	24/06/2022	IC-1006-orx	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☐
109571	09/06/2022	C-1012-fm	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☐
112046	04/07/2022	C-10032-KK	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☑
113478	15/07/2022	C-10032-KK	[redacted]	[redacted]	[redacted]	0	0	05. Paid/Completed	☐
113511	20/07/2022	IC-1006-orx	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☐
				Non-chargeable	Why is App Testing a bit flaky in			03.	
13						1	0		

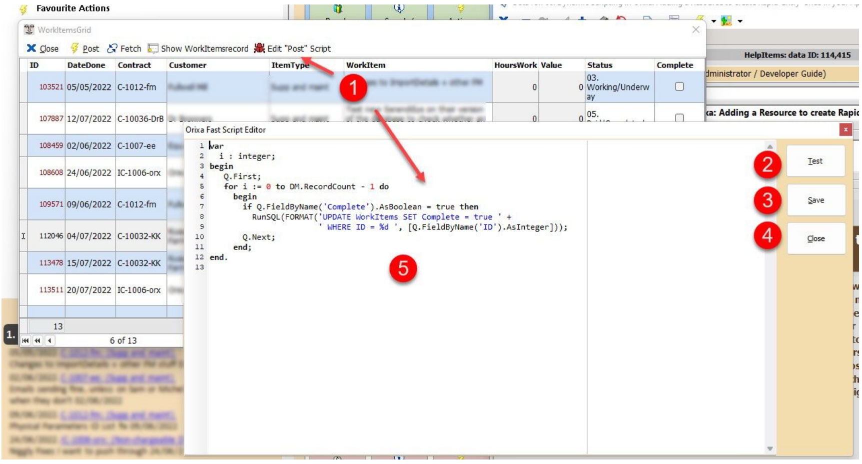
6 of 13

Rapid Entry Grid in Use

As soon as the Action is clicked, a data-grid form will open. The content of the data in this grid is set by the programming of the**Resource**.

> 1. The user can edit and update individual items of data in the grid, correcting and checking it as necessary.
> 2. Once **all rows** in the grid have been checked, the user can click "Post", doing this triggers the **pascal script** which can undertake any database actions decided on by the Developer.
> 3. If a row in the grid is linked to a record in the database, a "Show XXX Record" button will appear in the

3. rapid entry grid window. Clicking this opens the record linked to the currently selected row.
4. If a user has a high enough Security Level a button appears on the grid allowing them to edit the pascal script.



ID	DateDone	Contract	Customer	ItemType	WorkItem	HoursWork	Value	Status	Complete
103521	05/05/2022	C-1012-fm	[redacted]	[redacted]	[redacted]	0	0	03. Working/Underway	☐
107887	12/07/2022	C-10036-DrB	[redacted]	[redacted]	[redacted]	0	0	05.	☐
108459	02/06/2022	C-1007-ee	[redacted]						
108608	24/06/2022	IC-1006-orx	[redacted]						
109571	09/06/2022	C-1012-fm	[redacted]						
112046	04/07/2022	C-10032-KK	[redacted]						
113478	15/07/2022	C-10032-KK	[redacted]						
113511	20/07/2022	IC-1006-orx	[redacted]						

13

6 of 13

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

Editing the Pascal Scrpt of a Rapid Entry Grid

1. Click the "Edit Post Script" button. The Script Editing Window will appear showing the script that will run when the user clicks the "Post" button.
2. The Developer can click the "Test" button to**compile** the script. This checks that the script does not contain any syntax errors, and will run when the user clicks the "Post" button of the grid. If the script contains errors the compiler will return a message which should help to identify the error.
3. The Save button allows any changes made by the Developer to be recorded in the database, saving the script to the "ObjectProperties" field of the Resource which generates the grid.
4. The Close button closes the window and returns the user to the data-entry-grid screen.

A fuller tutorial about writing pascal-scripts in the script-editing-window is included here:

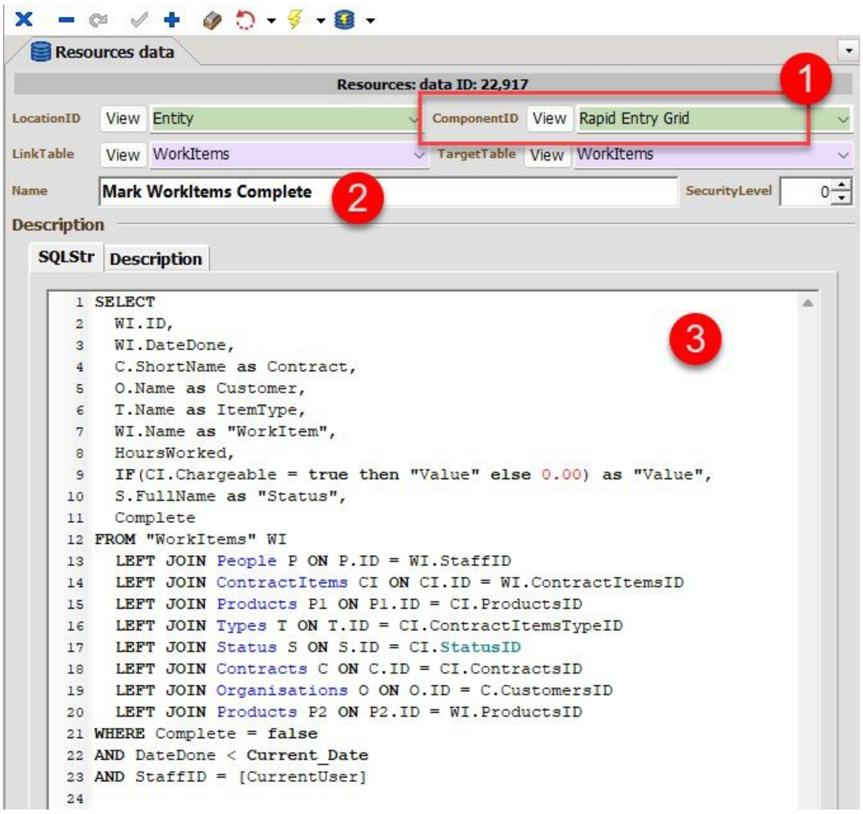
Introduction to Pascal Scripting in Orixa

Creating a Resource which is a Rapid Entry Grid

First create a Resource following the steps shown here:

Using the "Resources" system-table to create reports, charts, data-cubes and dashboards

Then reset the "ComponentID" of the record to "Rapid Entry Grid"



Resources: data ID: 22,917

LocationID: Entity

ComponentID: Rapid Entry Grid

LinkTable: WorkItems

TargetTable: WorkItems

Name: Mark WorkItems Complete

SecurityLevel: 0

Description

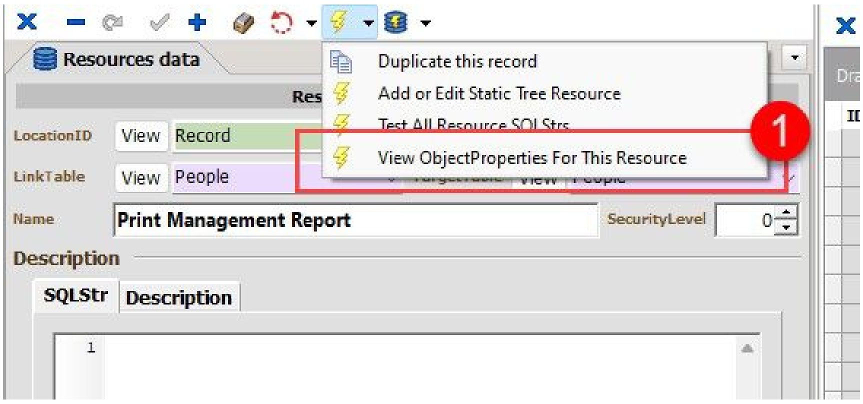
```
SELECT
  WI.ID,
  WI.DateDone,
  C.ShortName as Contract,
  O.Name as Customer,
  T.Name as ItemType,
  WI.Name as "WorkItem",
  HoursWorked,
  IF(CI.Chargeable = true then "Value" else 0.00) as "Value",
  S.FullName as "Status",
  Complete
FROM "WorkItems" WI
  LEFT JOIN People P ON P.ID = WI.StaffID
  LEFT JOIN ContractItems CI ON CI.ID = WI.ContractItemsID
  LEFT JOIN Products P1 ON P1.ID = CI.ProductsID
  LEFT JOIN Types T ON T.ID = CI.ContractItemsTypeID
  LEFT JOIN Status S ON S.ID = CI.StatusID
  LEFT JOIN Contracts C ON C.ID = CI.ContractsID
  LEFT JOIN Organisations O ON O.ID = C.CustomersID
  LEFT JOIN Products P2 ON P2.ID = WI.ProductsID
WHERE Complete = false
AND DateDone < Current_Date
AND StaffID = [CurrentUser]

```

Rapid Entry Grid Resource Record

> 1. Newly created resource with ComponentID = "Rapid Entry Grid", note that as the "LocationID" of this Resource is set to "Entity" it will appear in the Actions-menu of the WorkItems Entity on the System Entities screen.
> 2. The "Name" of the resource will be used to set the text used in the Actions-menu **and** the Caption of the ddata-grid window.
> 3. The SQL code for the resource will be used to populate the grid.

Viewing the Pascal Script that runs when the "Post" button is clicked by the user


Resources data
Res
Duplicate this record
Add or Edit Static Tree Resource
Test All Resource SQL Strs
View ObjectProperties For This Resource
1
SQLStr
Description
Dra
ID

LocationID: Record

LinkTable: People

Name: Print Management Report

SecurityLevel: 0

Description

Access the "ObjectProperties" of the Resource

The Pascal Script is stored in the "ObjectProperties" of the resource. Access this from the Actions menu of the Resources Edit Form. Details of how to write Pascal Scripts are covered in the Advance Developer sections of this help. For more information please refer there.