

Creating a new BusinessObject for an Orixas System

The process to do this is not simple, but if the steps are followed you should be able to add a functioning BusinessObject to your App. It is strongly recommended that any Admin or Developer user adding a new BusinessObject to their system first uses a test system and thoroughly tests any new systems prior to running them on their main system.

Once a BusinessObject has been created, the "CSV Importation Utility" can be used to import data from an external data-source, such as Excel, into your App.

Orixas has useful tools to make the creation of SQL Scripts needed to create BusinessObjects easier. However these cannot craft a full script in all cases. The Administrator will have to extend the script manually in these cases.

A BusinessObject consists of:

- A data-table in the database which has been defined following Orixas framework rules. This will have ID and DateCreated fields, plus any fields required for your data.
- A record added to the BusinessObjects table with "Name" matching the name of the data-table, and a "ViewScript". All other data in the BusinessObjects table is optional, but each field serves a purpose in your App, so review what you need to add. The BusinessObjects Record can be updated at any time, so it is not necessary for it to be completely correct from the outset.
- Records should also be added to the system-tables for "Searches", "Reports", "Resources", "Status" and "Types" to provide features for the BusinessObject. Consult Help topics on these system-tables to learn more about them.

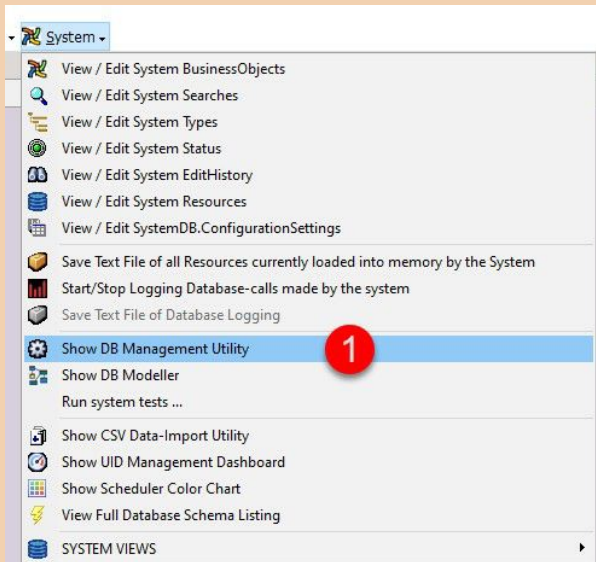
Initial Considerations

- Is the new BusinessObject a "child" or "extension" of one or more of the existing BusinessObjects in the App? If it is, the new BusinessObject should include Foreign Keys to its "Master" or "Parent" BusinessObject(s), and possibly also Decoration to control the child's behaviour and ensure it is displayed in the right way in the App.
- Do you definitely need to add a wholly new BusinessObject, or would it be better to extend the number of columns in an existing BusinessObject?
- Orixas has many standards for field-names and data-types. If you use the "Create BusinessObject" tool (detailed below) your data will automatically follow these rules.

Options for how to add a new BusinessObject

1. Use the "Create new Entity" tool in the DB Modeller to create a SQL script that will create the BusinessObject. This allows you to use a tool which ensures that the data-table structure will work within the Orixas system, and also adds several other useful steps, adding data to Orixas system tables that are needed by the BusinessObject. This will produce a rough SQL script, which will require some re-writing and extension. Once complete, if this script is run the new BusinessObject will be created
2. Find another BusinessObject (in your own App, or from the Orixas web-site) which is similar to the one you want to create and "reverse engineer" it. This will produce a SQL script, but several parts of it will not match your exact requirements, so you will need to change these to bring the script into line with your requirements
3. Hand-code a BusinessObject creation script: This is the hardest option, and requires a lot of skill, but once you have some template scripts to work with it is not so hard.

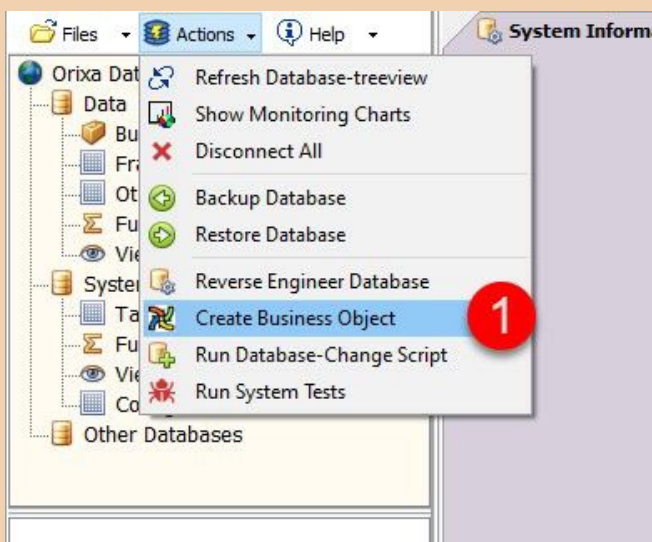
Accessing the "Create Business Object" window



System Menu, show DB Management Utility. Note that this menu is only visible if you have adequate privileges

The "Create Business Object" window is accessed from the "DB Management Utility", so first open this.

From the "System" menu, select "Show DB Management Utility".



Create Business Object

In the "DB Management Utility" select "Actions" "Create Business Object" (marked 1., in the image on the left).

The "New BusinessObject" screen

Note this is a complex screen, and it is very powerful. It allows the coder to carefully create the data-table following the exact standards required for the Orix framework.

This includes automatically coding for issues such as linking a data-table as a "child" of another, with foreign keys etc., adding "Type" and "Status" fields.

It also allows the creation of as many different Number, Date, String, and Linked fields as you want.

Virtually all the data requirements for any business object can be set using this screen.

If this screen is used, the resulting SQL script will be sure to follow Framework coding standards. It is always possible to use and extend this script with your own additions once you have the starting-point script.

Also, once a BusinessObject has been created it is always possible

New Table Object

New Object / Table Name: Generate Script Return to App

Field Creation Script

Parenting

☒ Standard UID - ID
☐ Is Extension
Master table:
☐ Is Child
Parent table:
Linking field name:

Date and Time fields

Commonly used:
☒ DateCreated ☐ DateStart
☐ DateDone ☐ DateEnd
Date fields (separate with comma):

Number and Currency fields

Commonly used:
☐ Value ☐ Quantity
Number "Float" fields (separate with comma):
Money "Decimal" fields (separate with comma):

Links To People / Staff

Commonly used: Automatically set as "CurrentUser"
☐ AuthorID ☐
☐ StaffID ☐
☐ StaffNotifyingID ☐
☐ StaffResponsibleID ☐

Other common fields

☐ Name ☐ Description
☐ OID ☐ Memo
☐ LinkTable/ID ☐ Comment

Look-up List fields

☐ TypeID ☐ StatusID
Linking field name(s) (separate with comma):
"ID List" field(s) (separate with comma):
"Distinct" field(s) (separate with comma):
Tables to link to via "ID" Field (use comma):

Tickbox fields

Commonly used:
☐ Current ☐ Complete
TickBox fields (separate with comma):

Create Business Object Window

to extend it at any time, so keep the initial definition as simple as possible, you can always add complexity and extend the features of the BusinessObject later.

New Table Object

New Object / Table Name: 1

Field Creation Script

Parenting

☒ Standard UID - ID 2
☐ Is Extension
Master table:
☐ Is Child 3
Parent table:
Linking field name:

Create Business Object: Table-Name

1. Set the Name of the data-table. The normal standard is to use only alphabetic characters (a-z), and to use capital letters to allow multi-word table-names. It is normal to use the plural, so a table holding details of a company's products is called "Products" not "Product". For example: Inspections, FarmFields, InspectionDetails, WorkItems, ChemicalTests.
2. Most tables are not directly related to one other table. In such cases the ID field is set using a UID function. In a few cases a table may be an "Extension" of another table. In such a case tick "Is Extension" and then select a table which will be the "Master" table. For example: People is a "master" table to the extension tables "Farmers" and "Staff". Data stored in "extension" tables is used to add data used only for some people. Not all people are farmers or staff, so it is useful to segregate the data in this way.
3. Many tables are "child" tables. For example the "WorkItems" data-table is a "child" of the "Work" data-table. This means that "WorkItems" will automatically display as a table linked to "Work" in a "one-to-many" relationship.

Date and Time fields

Commonly used:

☒ DateCreated ☐ DateStart

☐ DateDone ☐ DateEnd

Date fields (separate with comma)

Number and Currency fields

Commonly used:

☐ Value ☐ Quantity

Number "Float" fields (separate with comma)

Money "Decimal" fields (separate with comma)

Create Business Object: Date and Number fields

1. Add any date fields needed here. Orixa has built in systems to add framework elements when data-tables contain field-names "DateStart", "DateEnd", "DateDone" or "DateCreated."

Add names for any other date fields you might need in the empty data-entry fields.

If you tick "DateDone" the tool will automatically add a column with this name to the creation script as a "date" column.

2. Add "Floating point number" fields (ie data that needs to be recorded such as 12.5 or 103.77. And "Decimal" fields. Orixa uses the "Decimal" data-type for all columns that hold currency or money values.

If you tick "Value" or "Quantity" the tool will automatically add columns with these names and the correct data-types to the creation script.

Links To People / Staff

Commonly used:

☐ AuthorID ☐ Automatically set as "CurrentUser"

☐ StaffID ☐

☐ StaffNotifyingID ☐

☐ StaffResponsibleID ☐

Other common fields

☐ Name ☐ Description

☐ OID ☐ Memo

☐ LinkTable/ID ☐ Comment

Look-up List fields

☐ TypeID ☐ StatusID

Linking field name(s) (separate with comma)

Create Business Object: People-Links and other Common Field-names

1. Orixa has a number of automated systems to link records to people or staff. Tick these boxes to add fields with these names to the data-table. Tick the second tick-box if you want the field to automatically be set to the current user of the system when the data is edited.

2. Tick any of these boxes to add fields with these names to the data-table.

3. Orixa includes useful features to allow creation of "type" and "status" pick-lists. If you want these to be added fill in this part of the tool.

1. Orixa includes useful features to allow creation of ID-List pick-lists. These allow the user to select a set of entries from a list. If you want these to be added fill in this part of the tool.

"ID List" field(s) (separate with comma) 1

"Distinct" field(s) (separate with comma) 2

Tables to link to via "ID" Field (use comma) 3

Tickbox fields

Commonly used:

☐ Current ☐ Complete

TickBox fields (separate with comma) 4

Create Business Object: Other field-types

2. Orixia includes useful features to allow creation of "distinct" pick-lists. These allow the user to pick from lists of previously-used entries for a field. If you want these to be added fill in this part of the tool.
3. Orixia includes useful features to allow linking data-tables via ID fields. For example a "Purchases" record may be linked to a "Product" via a ProductsID field. If you want these to be added fill in this part of the tool. You simply need to add the table-names for every table you want to link. The tool will do the rest.
4. Add any "tick-box" (Boolean) fields you need. Orixia includes features relating to fields with the names "Current" and "Complete."

A completed "Create Business Object" Window

New Table Object

New Object / Table Name:

Field Creation
Script

Parenting

☒ Standard UID - ID
☐ Is Extension
Master table:

☒ Is Child
Parent table:

Linking field name

Date and Time fields

Commonly used:
☒ DateCreated ☐ DateStart
☒ DateDone ☐ DateEnd
Date fields (separate with comma)

Number and Currency fields

Commonly used:
☐ Value ☐ Quantity
Number "Float" fields (separate with comma)

Money "Decimal" fields (separate with comma)

Links To People / Staff

Commonly used:
☒ AuthorID
☒ StaffID
☐ StaffNotifyingID
☐ StaffResponsibleID

Automatically set as "CurrentUser"
☒
☐
☐

Other common fields

☐ Name ☐ Description
☐ OID ☐ Memo
☐ LinkTable/ID ☐ Comment

Look-up List fields

☒ TypeID ☐ StatusID
Linking field name(s) (separate with comma)

"ID List" field(s) (separate with comma)

"Distinct" field(s) (separate with comma)

Tables to link to via "ID" Field (use comma)

Tickbox fields

Commonly used:
☐ Current ☒ Complete
TickBox fields (separate with comma)

Create Business Object: Completed Entry

New Table Object

New Object / Table Name: **Generate Script** Return to App

Field Creation Script

Parenting

☒ Standard UID - ID
☐ Is Extension
 Master table:
☒ Is Child
 Parent table:
 Linking field name:

Date and Time fields

Commonly used:
☒ DateCreated ☐ DateStart
☒ DateDone ☐ DateEnd
 Date fields (separate with comma):

Number and Currency fields

Commonly used:
☐ Value ☐ Quantity
 Number "Float" fields (separate with comma):

 Money "Decimal" fields (separate with comma):

Links To People / Staff

Commonly used: Automatically set as "CurrentUser"
☒ AuthorID ☒
☒ StaffID ☒
☐ StaffNotifyingID ☐
☐ StaffResponsibleID ☐

Other common fields

☐ Name ☐ Description
☐ OID ☐ Memo
☐ LinkTable/ID ☐ Comment

Look-up List fields

☒ TypeID ☐ StatusID
 Linking field name(s) (separate with comma):

 "ID List" field(s) (separate with comma):

 "Distinct" field(s) (separate with comma):

 Tables to link to via "ID" Field (use comma):

Tickbox fields

Commonly used:
☐ Current ☒ Complete
 TickBox fields (separate with comma):

Create Business Object: Generate Script

Please take a moment to have a look at how it has been filled in.
 You can click the "Generate Script" button (marked "1." above) to generate the script.
 The script that is generated is shown directly below.

Note that the script includes not just the "CREATE" statement for the data-table in the database, but also adds scripts needed to add a record to the "BusinessObjects" system-table, which the App uses to show the new BusinessObject.

Also: It is important to point out that the auto-generated script is not fully complete. If the new BusinessObject includes a data-field called "Name", then the script will work well. However in many cases a "Name" field is not needed. In these cases the "DisplayScript" and "ListScript" of the BusinessObject will have to be re-crafted by hand.

If you are confused by the purpose of different parts of the BusinessObject Script. Please refer to this Help Document:

[The Purpose of BusinessObject Scripts](#)

The Script Generated by the "New BusinessObject" Tool

```
CREATE TABLE Purchases
(
  ID INTEGER DEFAULT UID() NOT NULL,
```

```

OrganisationsID INTEGER,
DateDone DATE DEFAULT Current_Date NOT NULL,
    ProductsID INTEGER,
DatePaid DATE,
PurchasesTypeID INTEGER,
QuantityOrdered FLOAT DEFAULT 0 NOT NULL,
ValuePaid DECIMAL(19,4) DEFAULT 0 NOT NULL,
QualityChecked BOOLEAN DEFAULT FALSE NOT NULL,
"Value" DECIMAL(19,4) DEFAULT 0 NOT NULL,
Quantity FLOAT DEFAULT 0 NOT NULL,
AuthorID INTEGER,
StaffID INTEGER DESCRIPTION 'CurrentUser',
DateCreated TIMESTAMP DEFAULT Current_Timestamp,
Complete BOOLEAN DEFAULT false NOT NULL,

CONSTRAINT "PK_Purchases" PRIMARY KEY ("ID"),
CONSTRAINT "OrganisationsID" FOREIGN KEY ("OrganisationsID")
REFERENCES "Organisations" ("ID") DESCRIPTION 'Parent' ,
CONSTRAINT "StaffID" FOREIGN KEY ("StaffID")
REFERENCES "Staff" ("ID"),
CONSTRAINT "PurchasesTypeID" FOREIGN KEY ("PurchasesTypeID")
REFERENCES "Types" ("ID"),
CONSTRAINT "ProductsID" FOREIGN KEY ("ProductsID")
REFERENCES "Products" ("ID")
)!
```

```
PUBLISH DATABASE "Data" TABLES "Purchases"!
```

```

CREATE INDEX "siDateCreated" ON "Purchases" ("DateCreated")!
CREATE INDEX "siDateDone" ON "Purchases" ("DateDone")!
```

```

INSERT INTO BusinessObjects
(Name, BusObjType, AddNewButton, Color, Icon,
LinkToImages, LinkToComments,
DisplayScript, ListScript,
ViewScript, DiaryScript,
SummaryScript)
VALUES
('Purchases', 'C', true, 16777215, 1,
false, false,
' SELECT
Name, ID
FROM "Purchases"
WHERE ID = %d ',
' SELECT
Name as FullName, ID
FROM "Purchases"
ORDER BY FullName ',
'SELECT
ID,
OrganisationsID,
DateDone,
DatePaid,
T.Name as PurchasesType,
P1.Name as Products,
QuantityOrdered,
ValuePaid,
QualityChecked,
"Value",
"Quantity",
Pe.FullName as Author,
Pe1.FullName as Staff,
DateCreated,
```



```

Complete
FROM "Purchases" P
LEFT JOIN Types T ON T.ID = P.PurchasesTypeID
LEFT JOIN OrganisationsID O ON P.OrganisationsIDID = OrganisationsID.ID
LEFT JOIN People Pe ON Pe.ID = P.AuthorID
LEFT JOIN People Pe2 ON Pe2.ID = P.StaffID
LEFT JOIN Products P1 ON P1.ID = P.ProductsID
%s ',
'',
''
)!
INSERT INTO Searches
(LinkTable, Name)
VALUES
('Purchases', 'All Records')!
INSERT INTO Searches
(LinkTable, Name, SQLStr)
VALUES
('Purchases', 'Recently Added',
' WHERE DateCreated > Current_Timestamp - INTERVAL '1' MONTH ')!
INSERT INTO Types
(LinkTable, LinkField, Name)
VALUES
('Purchases', 'PurchasesTypeID', '' )!

```

What is included and what is missed by the "New Business Object" Window?

All data-table creation is well-managed by the tool, so fields, foreign-key constraints, unique key generation etc., are all correct and do not need to be extended manually.

However, the tool cannot easily generate an accurate "ViewScript", "ListScript", "DisplayScript" or "DiaryScript" as exactly which fields are present in these scripts is highly dependent on user-needs. The tool generates a "bare bones" version of each Script, which a Developer will have to extend.

So, once you have the basic script, you should extend those parts of the definition that are incomplete, such as the "Display-Script", "Diary-Script" etc. The best / easiest way to do this is to use other Business Objects with similar properties as as basis. Refer to the "Purpose of BusinessObjects Scripts" help-topic mentioned above for more information.

Components of the creation script

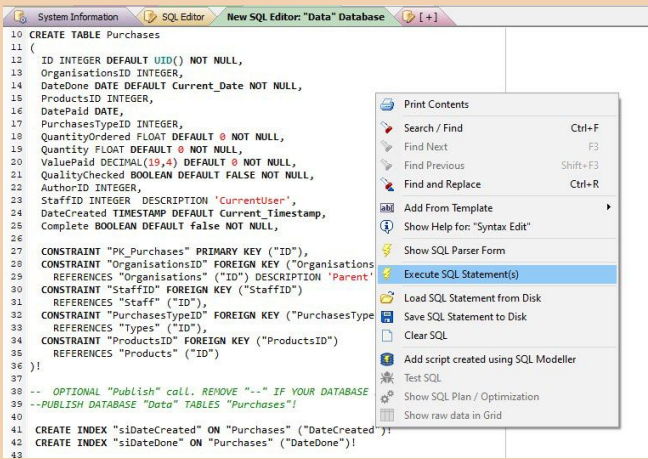
- A script to create the table, following Framework standards.
- A script to insert a row into the "BusinessObjects" data-table with a Name = the name of the new table, and entries for the different columns in the table to set behaviour as you wish it to function.
- The record should also have:
 - a "ViewScript" which is used to generate the "Grid-View" of a BusinessObject.
 - a "DisplayScript" and "ListScript" to allow the framework to generate lists of the records in the table.
 - a "DiaryScript" if the data-table contains date-based data and you wish it to be viewed in the system Diary.

Details of the operation of the BusinessObjects data-table are covered in more detail in other Help Documents.

Running the Script to create the new BusinessObjevct

Once you close the "New Entity Creation" screen, the script you have generated should be pasted into a script-viewer within your App, as shown.

The script can now be testing, rewritten and extended as needed. You can also save it to disk to use later, or to use as part of a longer upgrade process.



```
10 CREATE TABLE Purchases
11 (
12     ID INTEGER DEFAULT UID() NOT NULL,
13     OrganisationsID INTEGER,
14     DateDone DATE DEFAULT Current_Date NOT NULL,
15     ProductsID INTEGER,
16     DatePaid DATE,
17     PurchasesTypeID INTEGER,
18     QuantityOrdered FLOAT DEFAULT 0 NOT NULL,
19     Quantity FLOAT DEFAULT 0 NOT NULL,
20     ValuePaid DECIMAL(19,4) DEFAULT 0 NOT NULL,
21     QualityChecked BOOLEAN DEFAULT FALSE NOT NULL,
22     AuthorID INTEGER,
23     StaffID INTEGER DESCRIPTION 'CurrentUser',
24     DateCreated TIMESTAMP DEFAULT Current_Timestamp,
25     Complete BOOLEAN DEFAULT false NOT NULL,
26
27     CONSTRAINT "PK_Purchases" PRIMARY KEY ("ID"),
28     CONSTRAINT "OrganisationsID" FOREIGN KEY ("OrganisationsID")
29     REFERENCES "Organisations" ("ID") DESCRIPTION 'Parent',
30     CONSTRAINT "StaffID" FOREIGN KEY ("StaffID")
31     REFERENCES "Staff" ("ID"),
32     CONSTRAINT "PurchasesTypeID" FOREIGN KEY ("PurchasesTypeID")
33     REFERENCES "Types" ("ID"),
34     CONSTRAINT "ProductsID" FOREIGN KEY ("ProductsID")
35     REFERENCES "Products" ("ID")
36
37 )!
38
39 -- OPTIONAL "Publish" call. REMOVE "--" IF YOUR DATABASE
40 --PUBLISH DATABASE "Data" TABLES "Purchases"!
41
42 CREATE INDEX "sDateCreated" ON "Purchases" ("DateCreated")!
43 CREATE INDEX "sDateDone" ON "Purchases" ("DateDone")!
```

Executing Business Object Creation Script

Sections of script are separated by exclamation marks.

As many parts of script as necessary can be run after each other.