

The Images System-table

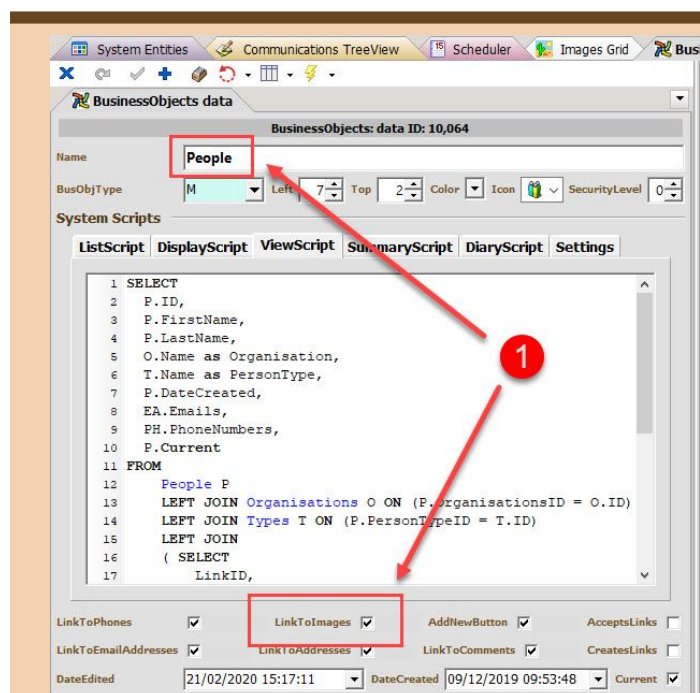
Orixa uses a single data-table to hold images that are linked to data-records. Developers can ignore this feature, and add fields to data-tables which store images if they wish.

The Images system-table provides a built-in mechanism for linking one or more images to any data-record in your App. These images might be pictures of a product or profile pictures of staff or users.

Once added images are viewed either with the data they are linked to, or via the Images Grid. Details of the Images Grid are in the following help-topic:

[The System Images Grid](#)

How Images are Linked to BusinessObjects in your App

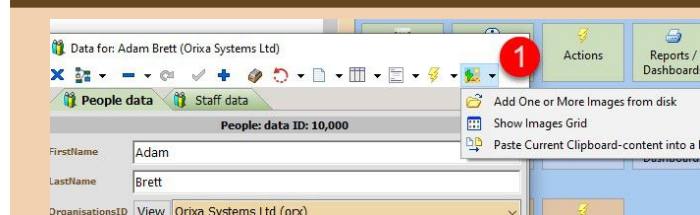


Images System Table Linking

Adding images-related features to one of your BusinessObjects is very simple.

In the BusinessObjects edit-form simply find the BusinessObject for which you wish to enable images, and tick the "LinkToImages" tick-box.

The next time you open your App and edit records in the selected BusinessObject, image-storage will be activated for this BusinessObject.



Images System Table Edit Form

"People" Edit Form with "LinkToImages" enabled

Once a BusinessObject is linked to Images, the "Images" icon will be added to the edit-form main toolbar (1).

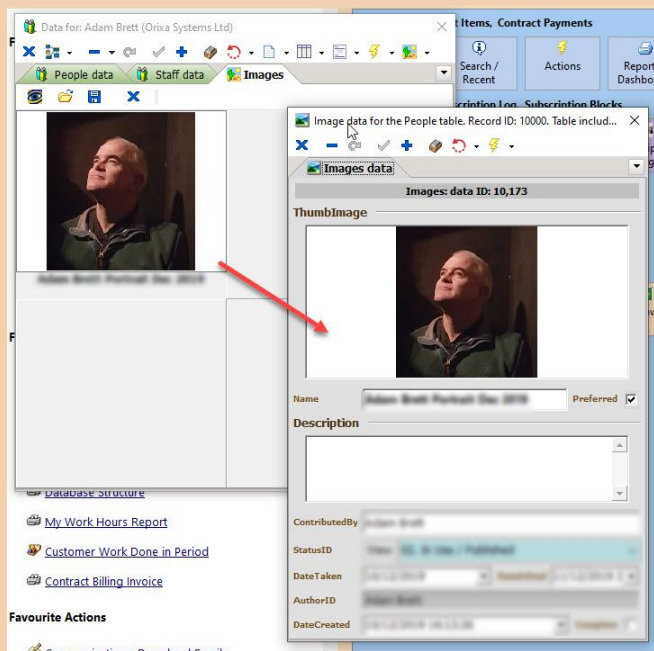
This allows you to add images which are linked to this data-record, view the images in an "Images Grid", and "Paste Current Clipboard-content into a new Image".

The "Paste Current" option in this menu is particularly useful if users wish to for example take "screen grabs" and add these to the database.

Images Grid for individual person

If Show Images Grid is selected a tab is added in the edit-form and linked images are shown (the grid may be empty if no records are present).

From here it is possible to open the Edit Form for each image, if required (as shown on the left)



Images System Table Edit Form Grid

How Images can be used in Orixasoft Apps

All Orixasoft systems can process and display the images that are stored in the images system-table.

Therefore it is easy to use these images in any of the user-interface parts of your App. For example, product photos can be added to reports, dashboards and dataviewers that are output from your App.

Simply add a reference to the Images system-table in your SQL statement.

It is important to note that the Images system-table includes 3 image-fields which are automatically created when you import image data. These are the "Image", "MediumImage" and "ThumbImage" fields. These exist to make it easy to retrieve just a small ("Thumb") or medium-sized version of the image if you do not a full sized image. As images are very data-intensive this helps to keep the operation of Orixasoft Apps snappy and fast. The "Image" data holds the original, "MediumImage" holds a version which is 500 pixels wide, and "ThumbImage" holds a version which is 150 pixels wide.

```

1 SELECT
2 P.ID,
3 P.FullName as Product,
4 I.Name as ImageDetails,
5 I.ThumbImage,
6 T.Name as ProductType,
7 S.FullName as "Status"
8 FROM
9     Products P
10    LEFT JOIN Images I ON (P.ID=I.LinkID)
11    LEFT JOIN Types T ON (P.ProductsTypeID=T.ID)
12    LEFT JOIN Status S ON (P.StatusID=S.ID)
13 WHERE P.ProductsTypeID = [LUTypes ProductsTypeID, Products]
14 AND P.StatusID = [LUStatus Products]
15 AND I.Preferred = true
16
17 ORDER BY P.FullName

```

SQL to Add Images to a Report

Note: The SQL JOIN is made between the "ID" field of "Products" and the "LinkID" field of Images.

SQL to return a list of products with a thumb-nail of each product's "preferred" image

```

SELECT
    P.ID,
    P.FullName as Product,
    I.Name as ImageDetails,
    I.ThumbImage,
    T.Name as ProductType,
    S.FullName as "Status"
FROM Products P
LEFT JOIN Images I ON (P.ID=I.LinkID)
LEFT JOIN Types T ON (P.ProductsTypeID=T.ID)
LEFT JOIN Status S ON (P.StatusID=S.ID) WHERE
P.ProductsTypeID = [LUTypes ProductsTypeID,
Products]
AND P.StatusID = [LUStatus Products]
AND I.Preferred = true
ORDER BY P.FullName

```

```

1 -- "Images" SQL DEFINITION --
2 CREATE TABLE "Images"
3 (
4 "ID" INTEGER DEFAULT UID() NOT NULL,
5 "LinkTable" VARCHAR(100) COLLATE "ANSI",
6 "LinkID" INTEGER,
7 "Name" VARCHAR(70) COLLATE "ANSI",
8 "ThumbImage" BLOB,
9 "MediumImage" BLOB,
10 "Image" BLOB,
11 "FileName" VARCHAR(120) COLLATE "ANSI",
12 "Description" CLOB COLLATE "ANSI",
13 "StatusID" INTEGER DEFAULT StatusID('In Use / Published'),
14 "Preferred" BOOLEAN DEFAULT false,
15 "Size" INTEGER,
16 "ContributedBy" VARCHAR(40) COLLATE "ANSI",
17 "DateTaken" DATE,
18 "DateEdited" TIMESTAMP DEFAULT Current_Date,
19 "AuthorID" INTEGER,
20 "DateCreated" TIMESTAMP DEFAULT Current_Timestamp,
21 "Complete" BOOLEAN DEFAULT false,
22 CONSTRAINT "PK_Images" PRIMARY KEY ("ID"),
23 CONSTRAINT "StatusID" FOREIGN KEY ("StatusID") REFERENCES "Status" ("ID")
24 ON UPDATE NO ACTION ON DELETE NO ACTION
25 )

```

Images SQL Definition

Note that Orixia uses 3 fields to hold the image data with different levels of definition. The "Image" data holds the original, "MediumImage" holds a version which is 500 pixels wide, and "ThumbImage" holds a version which is 150 pixels wide.

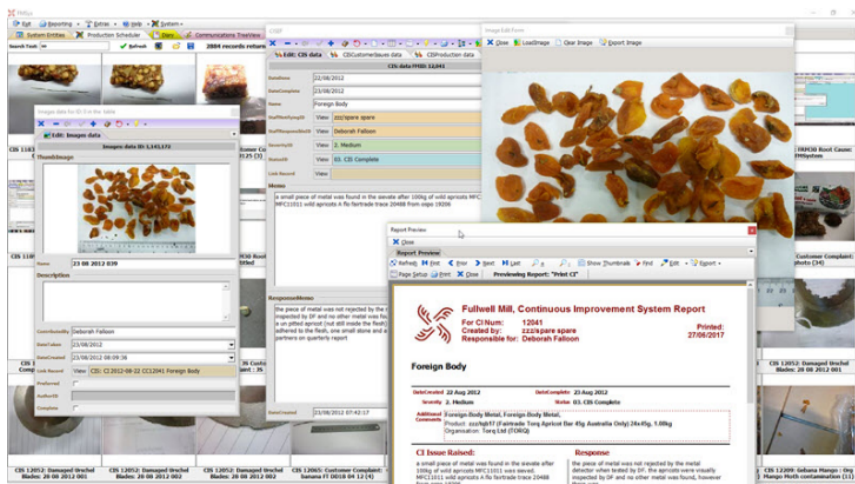
The SQL definition for the Images system-table

```

CREATE TABLE "Images"
( "ID" INTEGER DEFAULT UID() NOT NULL,
"LinkTable" VARCHAR(100) COLLATE "ANSI",
"LinkID" INTEGER, "Name" VARCHAR(70) COLLATE
"ANSI",
"ThumbImage" BLOB,
"MediumImage" BLOB,
"Image" BLOB,
"FileName" VARCHAR(120) COLLATE "ANSI",
"Description" CLOB COLLATE "ANSI",
"StatusID" INTEGER DEFAULT StatusID('In Use /
Published'),
"Preferred" BOOLEAN DEFAULT false,
"Size" INTEGER,
"ContributedBy" VARCHAR(40) COLLATE "ANSI",
"DateTaken" DATE,
"DateEdited" TIMESTAMP DEFAULT Current_Date,
"AuthorID" INTEGER,
"DateCreated" TIMESTAMP DEFAULT
Current_Timestamp, "Complete" BOOLEAN DEFAULT
false,
CONSTRAINT "PK_Images" PRIMARY KEY ("ID"),
CONSTRAINT "StatusID" FOREIGN KEY
("StatusID") REFERENCES "Status" ("ID")
)

```

An image in-use in an App Report



An Enterprise-quality-system App, which adds regularly captured images of food products for personnel to monitor and review.