

# Adding "PDFRunner" and "EmailRunner" Dynamic Link Libraries(DLLs) to a PDF creation and email in your Orixa App

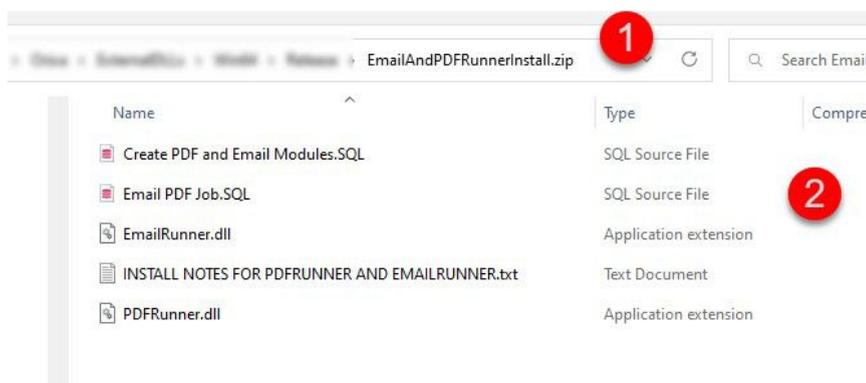
It is often useful to create Adobe Acrobat PDF files based on the contents of your Orixa App database. For example a PDF might show customer sales or a management finance report.

It can often be useful to send an email to selected people automatically without human intervention. For example an email saying that a backup process has completed, or some other database activity has occurred.

The two DLLs can be used together: A PDF File can be created by a procedure, and then emailed using the second procedure. These automation processes can greatly save staff time and effort in otherwise cumbersome activities.

Orixa includes Dynamic Link Libraries (DLLs) which enable this behaviour. Adding this behaviour to your App is somewhat technical, but if you follow the steps outlined here is not too difficult.

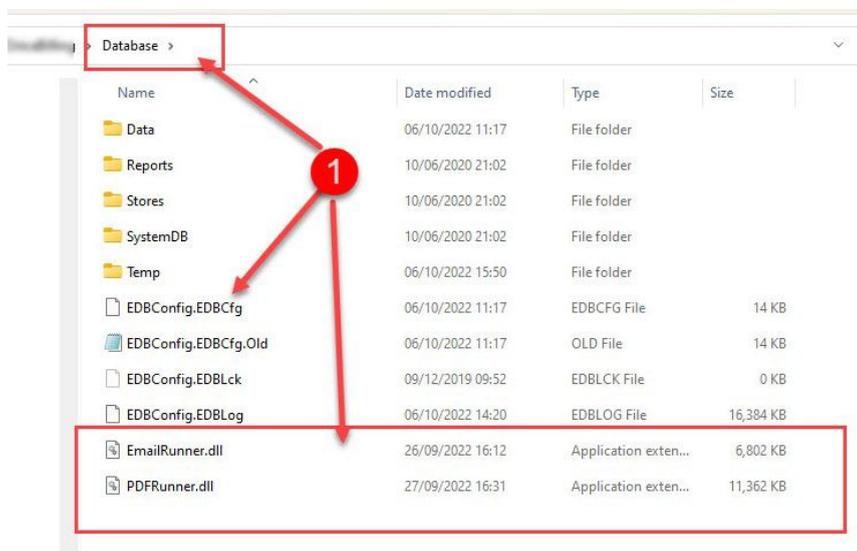
## Using the PDFRunner and EmailRunner DLLs



Contents of the "EmailAndPDFRunnerInstall.zip"

1. Download the DLL(s) you need from the Orixa Downloads page.
2. Open it using your Orixa password. You should see 2 SQL files which will alter your database to add the functionality, 2 DLL files which actually do the work, and an install notes file.

## Next Steps



DLL's in their destination folder

- Copy the DLL(s) into the same folder on your database server computer as your "EDBConfig.EDBCfg" file, as shown in the image above. Usually this is a folder called "Database" which contains all database-related files.
- In your App open the DB management Utility and run the "Create PDF and Email Modules.SQL" this will create 3 stored-procedures: EmailRunner, PDFRunner and SaveEmailSettings.
- Run the SaveEmailSettings procedure using your own email-server details.
- Now when EmailRunner and PDFRunner are called the parameters entered into them will be passed to the DLL(s) and should result in the creation of PDF files and / or sending of emails.
- The final step is to create JOBS and PROCEDURES which call the EmailRunner and PDFRunner.

## SQL to Create PDFRunner and SQLRunner Stored Procedures

```

CREATE MODULE PDFRunner
PATH 'C:\[Your App Folder]\Database\PDFRunner.dll'
!
CREATE MODULE EmailRunner
PATH 'C:\[Your App FolderYour App Folder]\Database\EmailRunner.dll'
!
CREATE PROCEDURE PDFRunner(IN "AppName" VARCHAR,
                           IN "ResourceID" INTEGER,
                           IN "FileName" VARCHAR,
                           OUT "Result" VARCHAR)

EXTERNAL NAME PDFRunner
!
CREATE PROCEDURE "EmailRunner" (IN "To" VARCHAR,
                                IN "Subject" VARCHAR,
                                IN "Body" VARCHAR,
                                IN "Attachments" VARCHAR,
                                OUT "Result" VARCHAR)

EXTERNAL Name EmailRunner
!
CREATE PROCEDURE "SaveEmailSettings" (IN "Host" VARCHAR,
                                       IN "Password" VARCHAR,
                                       IN "FromAddress" VARCHAR,
                                       IN "Port" INTEGER,
                                       IN AuthenticationType INTEGER,
                                       OUT "Result" VARCHAR)

EXTERNAL Name EmailRunner
!

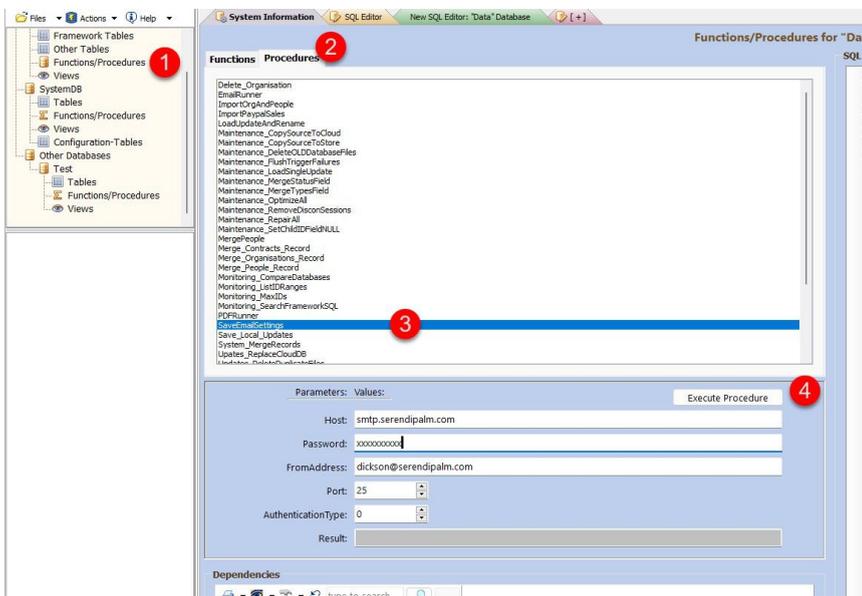
```

### NOTES:

**You will have to refresh the Database Management Utility Tree-view after this script is run to ensure that the new procedures are added to the view.**

**The SaveEmailSettings procedure encrypts your email-password to avoid the risks of hacking.**

## How to Save Email Settings

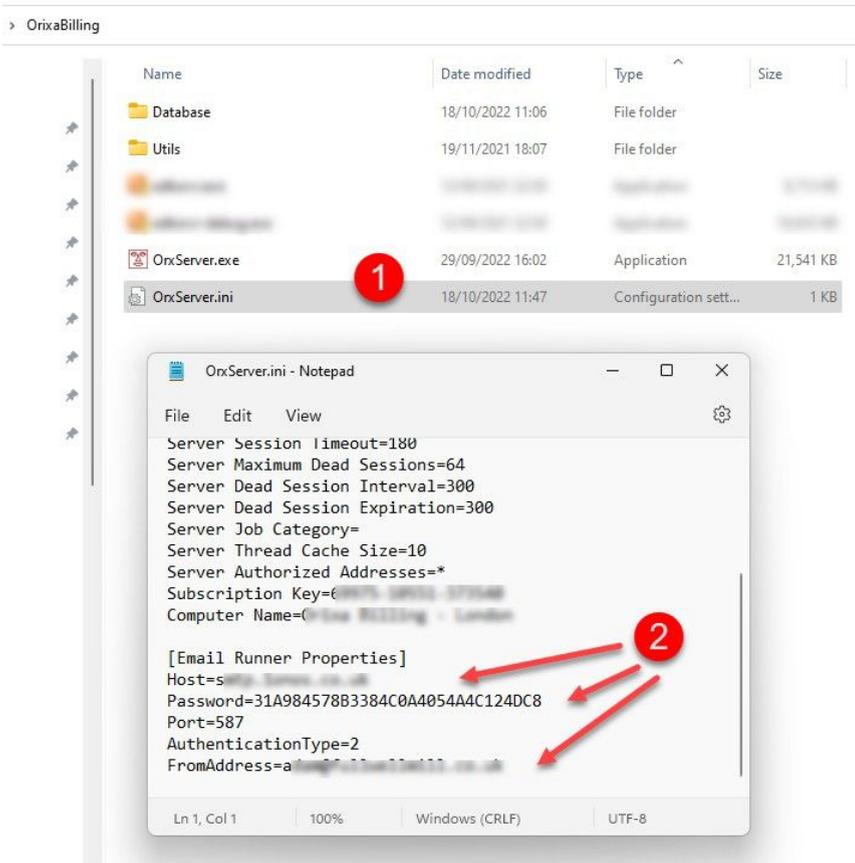


### Running Save Email Settings

1. In the Database management Utility, find the "Functions/Procedures" heading.
2. Click on the "Procedures" tab.
3. Click on the "SaveEmailSettings" procedure which should have been created when the "Create PDF and Email Modules.SQL" script was run.
4. Fill in your email-server settings, and click "Execute. The SMTP Host, Password, Account Email Address (FromAddress), Port and AuthenticationType are needed.  
The SMTP Host is usually something like "smtp.yourdomain.com"  
The Port is usually 587 or 25. Authentication type can be: 0 = None. 1 = Default. 2 = Secured.  
Most modern systems use the "Secured" (SASL) settings, so AuthenticationType should be set to "2".

**For assistance with these settings please consult the IT staff who set up the email systems on your computers. If no assistance is available you can save-settings and then attempt to send emails using different combinations of values.**

### Where are my Email Settings saved?

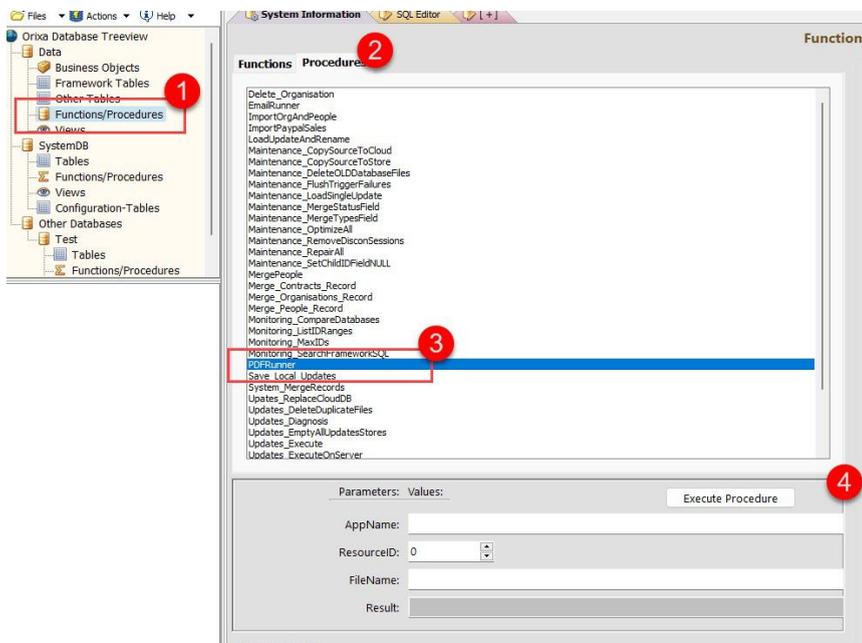


#### Email Settings Saved in OrxServer.ini

1. When the "SaveEmailSettings" procedure runs the settings you have entered are recorded in the "OrxServer.ini" file.
2. The settings are added as a new section to the INI file. Note that the password is encrypted, so it cannot easily be viewed / hacked.

## How to call the PDFRunner Stored Procedure from the DB Management Utility

The PDFRunner generates a PDF based on a Resource created in your Orixa App. Any Report or PDF File Resource can be used. If you want to create a wholly new PDF you must first create the Resource in your Orixa App for this new report, and then use the ID of this resource in the Stored Procedure. It is important to note that any Resource used for this purpose must include SQL which will run without user interaction. Special Orixa SQL syntax such as "MinDate" or "CurrentUser" cannot be present in the SQL used in this context.



### PDF Runner Stored Procedure

1. Find the "Functions/Procedures" heading of the Database Treeview
2. Click on the "Procedures" tab.
3. Click on the "PDFRunner" procedure
4. Complete the values needed for the procedure, and click "Execute Procedure"

**NOTE: In the current version of PDFRunner, each time you run the procedure you must click away from the procedure (3., in the image above) and click back onto it everytime you run it, or an error will occur. This is a known bug which should be fixed soon.**

### What values do I put in the boxes?

The parameters of the procedure exist to allow the DLL (which is a completely separate program from your Orixa App) to run. The "AppName" should be the name of the executable file of your Orixa App. The "ResourceID" should be the ID of a record from the Resources data-table that will produce the report to be sent. If you do not understand this step please refer to sections of Orixa Help relating to "Using the "Resources" System-table. The "FileName" should be an **output** filename. In otherwords this variable should be the name you want the PDF file to be called.

Note that if a file-name is created for a file that already exists, **the file will be overwritten**. Note that if a file-name is used but refers to a non-existent file-path, **the procedure may fail, depending on the set-up of your installation**. Bearing this in mind, set a file-name using an existing file-path. The Orixa standard behaviour is to save all these server generated files in the \\Database\Reports folder on the server, into suitably named sub-folders.

### Calling the PDFRunner stored procedure from a database JOB

**Database JOBS allow a stored procedure to be called at given intervals. This enables work to be done entirely automatically under the control of the Developer or Database-Admin staff.**

**The following SQL Creates a JOB to generate a PDF file with a specific name every day.**

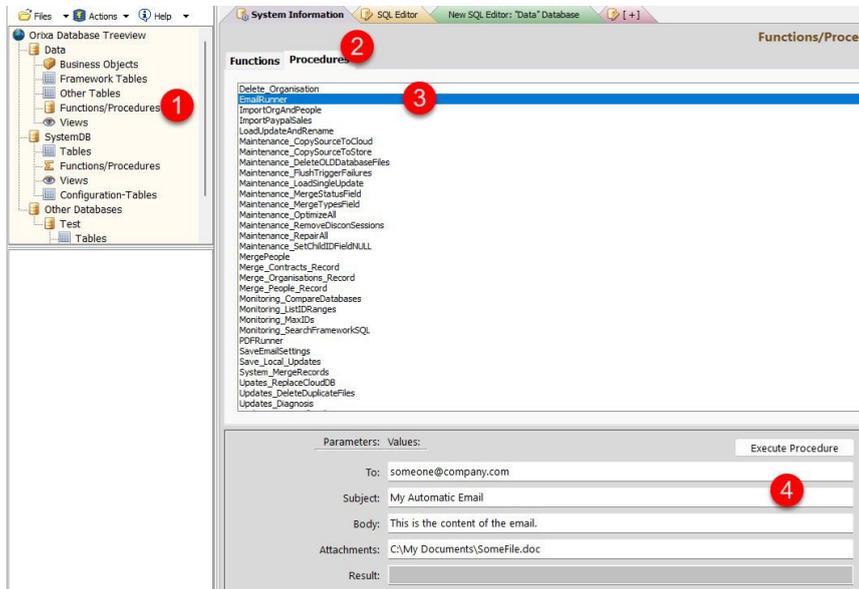
```
CREATE JOB "PDFRunnerMgtRep"
RUN AS "System"
FROM DATE '2021-12-09' TO DATE '2025-12-09'
DAILY ON MON, TUE, WED, THU, FRI, SAT, SUN
BETWEEN TIME '10:00' AND TIME '11:00:00.999'
CATEGORY 'System'
BEGIN
CALL PDFRunner('KKSys', 213342, 'C:\ServerDocs\ManagementFigures\MgtRep-' + CAST(Current_Date as
VARCHAR) + '.pdf');
END
```

Note how the above uses the "Current\_Date" SQL variable to provide part of the name for the PDF file. This ensures that each day a new file will be created, without one file over-writing another.

## How to call the EmailRunner Stored Procedure from the DB Management Utility

For the email runner to work all the data needed to create and send an email need to be passed to the EmailRunner.dll.

This means that the procedure needs to have seven oarameters passed in before it can work. The developer must know the name of the email host, and email password in order to create a JOB which references this stored procedure.



Accessing the EmailSender Procedure

1. Find "Functions/Procedures" in the Database Management Utility.
2. Click on the "Procedures" tab.
3. Click on the "EmailRunner" Procedure.
4. Complete the parameters of the procedure, and click "Execute Procedure".

### What values do I put in the boxes?

The parameters of the procedure are fairly self explanatory, you can see example values in the image above. The example which follows gives guidance on how to use the procedure.

## Calling the EmailRunner stored procedure from a database JOB

The following SQL Creates a JOB to send an email every day.

```
CREATE JOB "DailyEmailRunner"  
RUN AS "System"  
FROM DATE '2021-12-09' TO DATE '2025-12-09'  
DAILY ON MON, TUE, WED, THU, FRI, SAT, SUN  
BETWEEN TIME '10:00' AND TIME '11:00:00.999'  
CATEGORY 'System'  
BEGIN  
CALL EmailRunner('bethsmith@google.com', 'Test Email Subject Matter',  
                'Add email body here', 'c:\ServerDocs\SomeFileAttachment.txt' );  
END
```

The follow SQL Creates a PDF file and sends it in an email every day

```
CREATE JOB "EmailPDFMgtRep"  
RUN AS "Administrator"  
FROM DATE '2021-12-09' TO DATE '2025-12-09'
```

```

WEEKLY ON MON
BETWEEN TIME '10:00' AND TIME '11:00:00.999'
BEGIN
/*
SQL for a management report job.
I have added variables which are not strictly necessary, to make reading the
SQL easier and to make extending it easier, for example to send the same report to multiple
recipients with separate emails, you would just SET the aEmailAddress variable and CALL EmailRunner
a second time.
*/

DECLARE aFileName VARCHAR;
DECLARE aEmailAddress VARCHAR;
DECLARE aDatabaseName VARCHAR;
DECLARE aResourceID INTEGER;
DECLARE aEmailTitle VARCHAR;
DECLARE aEmailContent VARCHAR;

SET aFileName = 'C:\EDB\ORX\Database\Reports\ErrorLog ' + CAST(Current_Date as VARCHAR) + '.pdf';
    SET aEmailAddress = 'the email address';
SET aResourceID = 118394;
SET aDatabaseName= 'OrxWebServer';
SET aEmailTitle = 'Weekly ErrorLog Report ' + CAST(Current_Date as VARCHAR);
SET aEmailContent =
'Dear Adam,
Here is your ErrorLog Report.
Sent from the Orix AWS server';

USE "Data";
CALL PDFRunner(aDatabaseName, aResourceID, aFileName);
CALL EmailRunner(aEmailAddress, aEmailTitle,
aEmailContent, aFileName);
END

```