

FORMFUSION

FormFusion[®] ScriptDirector

Document version 1.9.0 – Unix/VMS/NT

Last Updated 4/17/2007

TABLE OF CONTENTS

OVERVIEW	3
FORMDIRECTOR: SCRIPTDIRECTOR.....	4
Break pages on	5
Binary format	5
Copy generated file to	5
Script	5
Storing files to the database	6
Storing files to a file system.....	6
ADDITIONAL DOCUMENTATION.....	7



OVERVIEW

FormFusion's new ScriptDirector, located and configured under the Script (PL/SQL) tab in FormDirector, will parse and convert .lis files into individual files.

ScriptDirector can be used with SunGard's new e-Bill functionality in Banner version 7.3. SunGard has enhanced both Student Self-Service Banner (SSB) and Accounts Receivable Internet Native Banner (INB) to allow students and back office staff to view the statement output of the Student Billing Statement process, TSRCBIL, when converted to PDF format and stored.

Specific instructions on how to use ScriptDirector to parse the .lis file and save the PDF files will be discussed in this document. Please refer to SunGard's AR 7.3 Release Guide and FAQs for additional instruction on the e-Bill functionality within Banner, including new table structures, Oracle directory creation and API specifics.



FORMDIRECTOR: SCRIPTDIRECTOR

The ScriptDirector module in FormDirector allows you to parse and store process output files into individual files. Depending on how the fields in ScriptDirector are populated, the individual files can be stored in the database, on the file system, or both.

The **Details** section in ScriptDirector contains 3 required fields, **Break pages on**, **Binary format** and **Script**, and one optional field, **Copy generated file to**.

FormDirector Properties

Select a copy and then configure the FormDirectors that will execute for that copy

Choose the copy for which to alter FormDirectors: Copy 1

Note: Copies are added and removed in the properties of the special print parameter

Archive | Email | Print | **Script (PL/SQL)** | Web

Execute this Director

Always Never Only if Archive does execute for page

Details

Break pages on: [] Binary format: None

Copy generated file to: []

Script: []

FormFusion

Commit Close Help

Break pages on

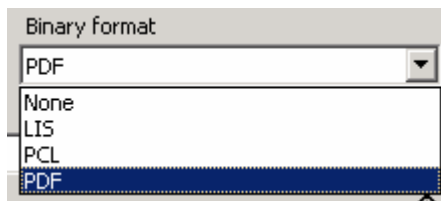


Break pages on
:stmt_number

A MapForm or CaptureForm variable should be entered with a colon preceding the variable name. This field determines how ScriptDirector will parse the output file into individual files.

In the example above, the stmt_number variable might be defined and populated in a MapForm offset field. The value used for this variable should be present on each page of the output file and positioned consistently from page to page.

Binary format



Binary format
PDF
None
LIS
PCL
PDF

Choose the binary format that ScriptDirector should use to create the files.

Copy generated file to

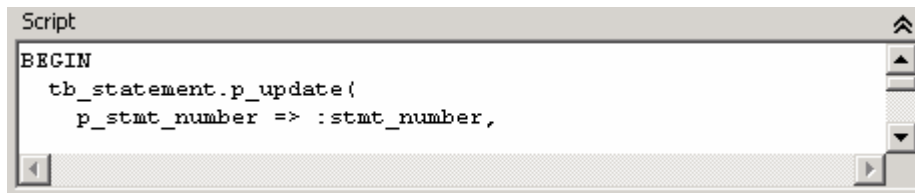


Copy generated file to
/my_server/ar_bills/:MyFileName

This optional field should be populated with a directory path when ScriptDirector is used to store each individual file into the file system.

In the example above, the MyFileName variable could be created in CaptureForm and populated with the statement number or any other data value unique to that file.

Script



Script
BEGIN
tb_statement.p_update(
p_stmt_number => :stmt_number,

PL/SQL scripting code should be entered to perform the actual processing of the output file parsing and storing.

The double-arrow symbol above the scroll bar can be selected to increase the size of the **Script** window. The other fields in the **Details** section will be hidden until the double-arrow is selected again.

Storing files to the database

Example script to store the individual files into the Banner database:

```
BEGIN
  tb_statement.p_update(
    p_stmt_number => :stmt_number,
    p_blot_code   => 'PDF',
    p_directory   => 'EVISIONS_DIR',
    p_filename    => :SCRIPTFILENAME_,
    p_data_origin => 'FormFusion');
  gb_common.p_commit();
END;
```

In this example, the FormFusion variable `stmt_number` would have previously been defined in MapForm or CaptureForm. The variable `:SCRIPTFILENAME_` is a pre-defined variable that holds the name of the temporary file created during processing.

Storing files to a file system

Example script to store the individual files on the file system:

```
BEGIN
  tb_statement.p_update(
    p_stmt_number => :stmt_number,
    p_blot_code   => 'PDF',
    p_directory   => 'EVISIONS_DIR',
    p_filename    => :MyFileName,
    p_loadfromfile => 'N',
    p_data_origin => 'FormFusion');
  gb_common.p_commit();
END;
```

In this example, the two FormFusion variables, `stmt_number` and `MyFileName` would have previously been defined in MapForm or CaptureForm.

In order to process the files, use `BANINST1` to create an Oracle directory logical that points to the folder where the files are located, and grant read access. If loading from the temporary directory used by job submission and Evisions in the `gjajobs` script, this might take the form:

```
create or replace directory EVISIONS_DIR as '/tmp';
create public synonym EVISIONS_DIR for EVISIONS_DIR;
grant read on directory EVISIONS_DIR to ban_default_m;
(or whatever role you used for FORMFUSION_SERVER access)
```

Note: It is important that the Oracle directory points to the same directory FormFusion stores its temporary files. If FormFusion stores the temporary files in `/jobsub/` and the Oracle directory is looking in `/tmp` you will get an error saying the pdf does not exist. You can specify which directory FormFusion should save its files to with the `-eviout` flag in `gjajobs.shl` (Unix), `gjajobs.com` (VMS), `sctban.pm(NT)`.



ADDITIONAL DOCUMENTATION

Please refer to SunGard's AR 7.3 Release Guide and FAQs for additional instruction on the e-Bill functionality within Banner, including new table structures, Oracle directory creation and API specifics.