

Datafile OS – Functionality, Configuration and Setup

Table of Contents

1. Datafile OS	2
1.1 Functionality.....	2
1.2 Running under Debian Linux.....	2
1.3 Access via Browser.....	2
1.4 Requirements for the User.....	2
1.5 Requirements for the System Supervisor.....	2
1.6 Requirements for the Installer.....	2
1.7 Requirements for the OS Developer.....	3
2. Raw Data Files	3
2.1 Obtaining EBCDIC PRCTL files.....	3
2.2 Files – Before and After Indexing.....	3
2.3 Configuration – dfsites.json and <sysid>.json file.....	4
3. Datafile PostgreSQL database	5
4. Datafile User Security	6
5. Indexing and Checking	6
5.1 Indexing.....	6
5.2 Checking the Index Run.....	7
5.3 Changing Parameters and Re-running.....	7
5.4 Preparing for Live Running.....	8
6. Printing	8

© Copyright 2021-2023 – this document and others associated with Datafile OS and produced by Project Computers Ltd. are the property of Project Computers Ltd. In the spirit of Open Source, you are welcome to use or copy material but please acknowledge the copyright.

Datafile OS – Functionality, Configuration and Setup

1. Datafile OS

1.1 Functionality

Datafile OS allows users to access reports produced elsewhere (e.g. IBM AS400) through a browser (subject to login constraints) to :

- ✓ view, page and search
- ✓ print
- ✓ cut-and-paste

1.2 Running under Debian Linux

Datafile OS is a Linux program suite under the [GNU v3 licence](#) supported by the Free Software Foundation. It's functionality extends only to certain report types but is freely available for you to modify or extend. At the moment it supports EBCDIC PRCTL files from the IBM AS400 range, and its successors.

Datafile OS runs under Debian Linux (other Linux versions may be available – see our web site [Project Computers Limited](#) for details on Datafile OS).

1.3 Access via Browser

Reports may be viewed, searched, printed etc. by any authorised user using an ACID-compliant browser. Only the most recent browsers are supported.

1.4 Requirements for the User

Users require only a browser (see 1.3) and authorised access to read the reports from the Datafile Linux server.

1.5 Requirements for the System Supervisor

The system supervisor, or operators, will require limited Linux knowledge and an ability to run command line tasks.

1.6 Requirements for the Installer

The Datafile OS installer will need a Debian 11 Linux server or know how to install one.

Scripts which install Datafile OS and associated packages are available from the web site which make the installation process fairly painless. A text editor is required to configure certain files.

If you wish to install on versions of Linux not supported by script, you will require fairly extensive of Linux and Linux tools including Apache 2, PostgreSQL (plus either phppgadmin or psql).

Scripts and documentation to accompany the software are included in the download to assist in the installation process. Notes on the indexing process and how to configure/test are available in Section 5 below.

Datafile OS – Functionality, Configuration and Setup

1.7 Requirements for the OS Developer

In addition to the requirements for an installer, any developer will need programming knowledge of C++, PHP, Javascript and HTML/CSS.

2. Raw Data Files

Datafile OS is currently available to process AS400 (or successors) report files in EBCDIC PRCTL format. Any OS developers are welcome to modify the system to accommodate other file formats e.g. ASCII, AS400 FCFC etc.

2.1 Obtaining EBCDIC PRCTL files

These files will need to be transferred from an AS400 (or its successors) – commonly using .TTO files. The transfer process is outside the scope of DatafileOS which expects the files to be presented in the Data Directory (as specified in the .json file – see 2.3 below).

Each separate report grouping or type should be presented in a separate physical file. For example, if statements are in a different format from the majority of the reports these should be downloaded in a separate named file. All files must have a suffix indicating line length, including the line leader (as in the 4-byte leader for PRCTL).

If the reports, with the exception of such items as statements, confirmations and advices, are in a standard format, with the same line length and with the same header structure (allowing the identification of the Report Identifier) then these should all be placed in a single bucket file called DOWNLOAD.136.

So your downloaded files on the Datafile server might include :

- ✓ DOWNLOAD.136
- ✓ STATEMENTS.136
- ✓ ADVICES.136

2.2 Files – Before and After Indexing

The files to be indexed (see 2.1 above) should be placed in a directory with the <system id> as the directory name. The Datafile OS system can be configured to run more than one report capture/retrieval system.

So if your system is Banking then you might want to have a 3-character system id – BNK

Datafile OS – Functionality, Configuration and Setup

2.3 Configuration – dfsites.json and <sysid>.json file

The file dfsites.json will contain the identifiers for any sites you wish to process. You can configure more than one – supposing you have reports from, say a banking **and** a trading system. These identifiers are listed in in dfsites.json as shown below :

```
{
  "sites": [
    "fin",
    "mis"
  ]
}
```

The system can be specified when indexing or chosen by the end user from a drop-down on the Report Search page.

Each system requires a .json file (one per source system) which must be present in the PCLDAT directory beneath /home/datafile. e.g. bnk.json.

It should contain :

Datafile OS – Functionality, Configuration and Setup

```
{
  "dataDir": "/home/datafile/bnk",
  "leaderLength": "4",
  "searchStart": "100",
  "mrlLen": "12",
  "recLen": 136,
  "jump": [
    15,
    15
  ],
  "searches": [
    "FINASTRA",
    "MISYS"
  ]
}
```

where :

dataDir	Beneath the home path of the special Linux user datafile a directory for raw and indexed data
leaderLength	Only for formatted systems e.g. EBCDIC PRCTL where there is a 4-byte line leader which specifies the formatted
searchStart	Column position at which to start the search for the “searches”
mrlLen	Specifically for reports where there is text to jump over between the search string – e.g. FINASTRA – and the Report Identifier
recLen	Total number of columns per line, including the leaderLength e.g. 132 column report plus 4 byte leader
jump	No. of characters from the start of the “searches” string to jump to the report identifier
searches	One or more text strings which precedes the Report Identifier

Note that a number of these parameters will not apply to files downloaded with their own name e.g. STATEMENTS.136 where STATEMENTS will be the report identifier used. The same would also be true if the system were modified for ASCII report files where typically each file represents a single report. File names and report identifiers are case-sensitive.

3. Datafile PostgreSQL database

The separate documentation/script used to install the Datafile OS system contains details on the database name **datafile** and the creation of the Linux user **datafile**.

Datafile OS – Functionality, Configuration and Setup

The database contains the following tables :

df_mainindex	One row per report identifier found containing report identifier, index date, total pages etc.
df_mainindex_def	One row per index run with system id, index date and information etc.
pcl_groups	<i>Not in use in this version</i>
pcl_groupusers	<i>Not in use in this version</i>
pcl_users	See Section 4 – you and the users will require a Datafile user to be created in order to log into the system

Note that only indices and security information are stored in the PostgreSQL database. The reports remain external to the database to satisfy the security and audit requirements of our customers. Data files are held in their original format, exactly as they were downloaded from the host machine.

4. Datafile User Security

Each user wishing to use the browser Datafile OS should have a unique User Id and password which will be required on the login screen before access to the system and reports is granted.

Indexing is carried out by the Linux user **datafile** from the command line, and utilises the strong security provided by Debian Linux.

5. Indexing and Checking

5.1 Indexing

The files to be indexed (see 2.1 above) should be placed in a directory with the <system id> as the name. Having ensured all the raw data files for the indexing date are present in the correct directory (e.g. *home/datafile/bnk*) you should (as the **datafile** Linux user) login and from the home path (~) of the datafile user, run the indexing as follows :

```
indexing <systemid> <date (format yyyyymmdd)>
```

e.g. indexing bnk 20220924

This should display a line for each report located, with the total pages found and the start and end offsets within file.

When completed, you should use the browser Datafile OS to view and check reports for the date entered.

Datafile OS – Functionality, Configuration and Setup

5.2 Checking the Index Run

Browse to the Datafile OS system : <http://www.datafile.sys>

Note that the online HTML help provides details of how to log in and recall reports so this will not be covered here.

Login with the Datafile user id and password created in the PostgreSQL database.

Hit the 🔍 search button to enter the Report Retrieval program.

To check all reports for the date processed, leave the Report Id. Field blank, type the date (YYYYMMDD) into the Start Date field, hit tab to copy it to the End Date field and then hit the 🔍 search button.

You should see a list of reports (one per line) showing the Report Identifier, Index Date and Total Pages. Click on any row to view the report. We recommend checking all reports (especially top and bottom of each report) to ensure that it has found the report correctly.

If you find that reports have not paged correctly or report identifiers have not been located then compare the data with the parameters set in the <sysid>.json file.

If you find that there are report entries with a report identifier of DUFF (Don't Understand File Format) then you need to view the report and see if you can identify the cause.

Common causes of DUFF entries are :

- x Banner page – if possible, ensure that banner pages are excluded from the downloaded reports
- x Different format from that of the rest of the reports e.g. including statements in the DOWNLOAD.136 file – these should be downloaded as a separate file e.g. STATEMENTS.136
- x Failure to locate the Report Identifier – perhaps the system name has changed (e.g. KAPITI to MISYS) – ensure that the appropriate search is included in the <sysid>.json file

If the whole DOWNLOAD.136 file is represented as a single DUFF entry then your <sysid>.json file contains a serious fault) often simple to fix. Make sure that the correct search(es) are present, line length and leader length are correct, also the MRL len and the start position.

5.3 Changing Parameters and Re-running

If you are testing the indexing process, checking the reports, changing parameters and then re-running the simplest method is :

- ✓ Rename the indexed files back to the original download file (e.g. 35012.DAT to DOWNLOAD.136)
- ✓ Use a PostgreSQL utility – phpPgadmin from the browser or psql from the command line - to clear the relevant items from the df_mainindex table

Datafile OS – Functionality, Configuration and Setup

5.4 Preparing for Live Running

Once you are satisfied that your reports are being captured and indexed successfully you should :

- ✓ Use a PostgreSQL utility – phppgadmin from the browser or psql from the command line to empty the contents of the df_mainindex **and** the df_mainindex_def tables
- ✓ It is also advisable to use an editor to reset the Next File Number to 00001 in */home/datafile/pcldat/nextfileno*

6. Printing

Printing reports which have a fixed format and fixed font may need some parameters to ensure that , for example, a 132 column report is printed in landscape and using a fixed font (or the columns will be haywire) and one of a size to fit into the landscape page.

A default and a report example are set up in the sample fin.json file under the printing section

```
"printing": [{
  "default": [{
    "font": "Courier",
    "size": 6,
    "orientation": "P",
    "paperSize": "A4"
  }]
}]
```

These settings can be tested using the Print options when viewing a report. If you need to adjust the default settings, or to set up an entry specifically for a report id. This can be done by editing the fin.json file as above.

The fonts available are from the fpdf library for php – they are specified as font families – look in the directory */var/www/datafile.sys/public.html/fpdf*.