

FDRWEB...Disk & Tape Reports with JUST 3 CLICKS



1st CLICK

2nd CLICK

3rd CLICK

NEW PRODUCT

EASY-TO-USE log on to FDRWEB with your TSO userid authorization. **RUNNING** as a USS application. **READY to RUN** hundreds of built-in reports. **IMPORTS**

Existing FDREPORT jobs on the fly. **REPORTING** generate Standard, Custom and CSV Formats which can be emailed. **MANAGE** your disk and storage operations with less resources. With a No Obligation 90-Day FREE Trial.



FDRWEB FREQUENTLY ASKED QUESTIONS

Below is a list of frequently asked questions by our customers. If you do not see the answer you are looking for please email: support@fdrinnovation.com

How are jobs submitted to the mainframe?

- ...Jobs are submitted to the mainframe using the internal reader assigned to the FDRWEB started task. Jobs can then be run on other LPARs using the ROUTE JCL command.

How does FDRWEB handle passwords and authentication for z/OS?

- ...In order to log in to FDRWEB, a user simply enters their existing TSO ID and password. FDRWEB then uses RACF/SAF authentication to verify the users credentials. We plan to provide support for pass phrases, as well as PassTicket in the future.

How does FDRWEB send emails?

- ...In order to send emails, FDRWEB requires a dedicated email address registered to your email service. In order to set up email capabilities, the ADMIN will need to provide FDRWEB with the address of your SMTP server, and the email and password dedicated to the FDRWEB mail account. Once that is configured, FDRWEB will be able to connect directly to your SMTP server and email.

continued...



FDRWEB FAQs...continued

What is FDRWEB?

- ... FDRWEB is a Java-based service which provides users with an easy-to-use web interface for DASD report generation, and general JCL submission and job output retrieval. FDRWEB runs as a z/OS started task, residing in a USS filesystem, and is accessed via a web browser.

What language is FDRWEB written in?

- ... FDRWEB is written in multiple languages. The service running on z/OS is written primarily in Java, using C when necessary to communicate with z/OS. The web page user interface is written in HTML and Javascript.

What are the technical requirements to run FDRWEB?

- ... FDRWEB requires the following system specifications to run:
 - z/OS version 2.2 or higher • IBM z/OS Java 1.8 or higher • JES2 (not compatible with JES3)FDRWEB is compatible with any major web browser, with the exception of Internet Explorer. We recommend Firefox, Chrome or Edge. Versions of these browsers prior to 2016 are not compatible with FDRWEB.

How is FDRWEB installed?

- ... FDRWEB is installed using a shell script (the installation manager), which is executed by the administrator in a USS environment. The installation manager is an interactive terminal-based tool, which will prompt the administrator for necessary information, such as a PROCLIB to place the FDRWEB started task, HTTPS configuration, and the port to bind the web server to.

How do you configure FDRWEB?

- ... The first step in configuration is creating a dedicated FDRWEB user, which will run the FDRWEB started task. This user must have an OMVS segment and a TSO segment. Once FDRWEB is installed, and the started task is started, the administrator can log in to the FDRWEB webpage as an ADMIN user, where they can configure which LPARs FDRWEB can submit jobs to, the time zone of the system and set up the email configuration.

How does FDRWEB handle client/server communication?

- ... FDRWEB by default uses HTTP in order to transmit data between the mainframe and a users PC. Using the installation manager, an administrator can easily configure FDRWEB to run HTTPS using either a self-signed certificate or a certificate signed by an in-house certificate authority. We highly recommend configuring HTTPS upon installation to ensure the system is secured end-to-end. When HTTPS is enabled, data such as job output or passwords are encrypted over the wire, providing significantly better security compared to a typical 3270 emulator which sends plain text data back and forth to the mainframe.