

# Flux 8.0 Demonstration

The following describes a general demonstration of Flux 8.0.

1. **Setup**
  - a. Open browser tabs to the
    - i. [Console Page](#)
    - ii. [Monitor Page](#)
    - iii. [Example Workflows](#)
    - iv. [Log File Viewer](#)
  - b. Open the Flux Architecture Depiction in the Flux Solutions Architecture Document
2. **(Opening)** Flux orchestrates files. Our unique architecture integrates workflow, managed file transfer, monitoring, job scheduling, and error handling.
  - a. Distinguishing Features - Integration of workflow and managed file transfer. Entire interface is browser-based. Command-line and REST APIs. Triggers based on web service calls, database conditions, file existence and non-existence, and timer expressions.
3. **(Supported Platforms)** Sign on to the Flux Public Staging Server as s.flux.ly admin user
  - a. Windows, Linux, Unix, and AIX Support
  - b. Support for Java 6, 7, and starting with Flux 8.0.10 - Java 8. Will stop supporting Java 6 in Flux 8.0.11 (but only impacts the embedded web server)
  - c. Mention database support and application server support.
    - i. Flux comes with an embedded web application server. Also runs within Weblogic, Websphere, JBoss, and Tomcat.
    - ii. SQL Server, DB2, Oracle, Postgres, MySQL, and Derby
4. **(Architecture)** Review the general architecture of Flux - Engines (1 or many), Browsers, Workflows, and optionally, Agents. All of these components are managed through the Flux Repository.
5. **(Flux Repository)** Now on to the Repository tab
  - a. The Flux repository is where Flux workflows, encryption keys, engine and agent configurations, and business calendars are kept.
  - b. We will talk in the next section about those things that are not kept in the repository - such as the runtime configuration file.
  - c. The repository is shared amongst all the Flux engines in a Flux Cluster. Repository items can be downloaded from the Repository, uploaded from the repository, even promoted from one repository to another (such as from Development to QA to Production).
  - d. Workflows in a repository can be reused - e.g., the same workflow can be submitted to a Flux engine into different namespaces. In this manner, repository workflows can be workflow templates. But the name in the repository can be the same as when the workflow is submitted to an engine.
  - e. Of course, access to the repository and repository actions are controlled by Flux security.
  - f. We provide a number of example workflows for importing into your repository on our examples page.
6. **(Flux Workflows)** Let's now pick a Repository workflow and explore it's design
  - a. Select the /Paris/Billpay/ workflowFirst - Let's show the different kinds of Flux Triggers and Actions
    - i. What's a Flux Action and a Flux Trigger - you may ask ....
  - c. Go through Core, Database, Workflow, Java, Web Services, File, Enterprise Java, Notifications

Now - let's look at this workflow in detail. This workflow is an example we provided that uses actions and triggers from **Core**

    - i. Timer trigger
    - ii. Console Action

Workflow

    - i. For Each Element in a Collection

Database

    - i. Insert and Update

Java

    - i. Validate an item

File

    - i. Move to the Archive and Notification
    - ii. Mail notifications of errors
7. **(Flux Operations)** Present the Flux Operations Console - home page.
  - a. Role of the operations console.
  - b. Status and rows of the console.
  - c. Actions available from the console, e.g., pause, resume, expedite ...
  - d. Filtering on rows
    - i. By state (Firing, Waiting, ...)
    - ii. By Namespace (Delve into concept and uses of namespaces)
8. **(Flux Security)** Transition to the Security Tab
  - a. Mention Active Directory and LDAP integration that Flux provides - mapping an AD Role or Group to a Flux Role or Group
  - b. Briefly discuss engine, agents workflow tabs
  - c. On Repository and Workflows tab delve into controlling access by namespace
  - d. Then spend a few moments on the Operations Console Tab - that control the visibility of the tabs to Operations staff
9. **(Logs and Audit and Reports)** Review of Logs, Audit, and Run History Tab
10. **Concurrency and Clustering and Failover**
  - a. Runtime (or dynamic) configuration
  - b. Failover consideration
  - c. Restart and recovery and the Flux transaction (and transaction break)
11. **What's coming in Flux**
  - a. New features and enhancements. New monitor and updated designer
  - b. Provisioning and onboarding interactions
  - c. And of course, performance, reliability, and resiliency features.
12. **Questions?**



## Related articles

- [Executing SQL Commands Against the Embedded Derby Database](#)
- [Flux 8.0 Demonstration](#)