

Elf

Elf is a robust container designed to support scripted applications.

Refer to the [infrastructure diagram](#) and [interaction diagram](#) for where it fits into our general workflow-management scheme.

Elf's native scripting language is [Ogrescript](#), but it can be extended to run other kinds of scripts as well.

Downloads

Latest versions: [ELF-latest](#).

Elf Javadocs can be accessed at: [API](#).

Elf Core Goals

1. Robust
 - Must not fail unexpectedly
 - Must not fail silently
 - Must be able to determine cause of failure
2. Monitorable
 - Must report status information to listeners
 - Should report debug information to listeners
 - Should report progress information to listeners
 - Should standardize logging across scripting implementations
3. Extensible
 - Should allow for user extensions to core functionality without recompilation

Elf Core Functionality

Authentication Management

Individual sub-component may need to authenticate to services. The `AuthenticationManager` provides an extensible mechanism for managing authentication tokens.

Event Management

A centralized event-handling mechanism. Sub-components have a means of adding listeners for specific types of events as well as a standardized way to create and send events.

Identity Management

Components create an identity for their sub-components through this mechanism, which is also extensible.

Elf Container

Elf supports multiple types of containers.

Elf Core Components

ElfRuntime

Instantiated and started by the Eclipse platform. It is responsible for instantiation, management of lifetime (`configure`, `initialize`, `run`), and management of status (calling `Bootstrap.alive()`, `Bootstrap.failed()`, `Bootstrap.success()`) of the `Bootstrap` object.

This object does not throw any exceptions, but handles failures cleanly by returning an appropriate exit code.

Bootstrap

Responsible for parsing the command line arguments, the set-up of the runtime environment, the configuration of `ElfPlatform`, and the instantiation and management of the `Container`. The configuration of this class works with little mandatory configuration and fails cleanly if the container is missing or invalid.

ElfPlatform

The globally accessible point of access to Elf-wide objects. The various managers (Authentication, Event, and Identity) are instantiated and configured by this class from a file determined by a command line argument:

- `-bootstrap.properties <value>`.

The platform provides default managers if nothing specific is requested.

Elf Runtime Environment

The platform provides a standard environment for running scripted applications and a means of accessing these variables.

This environment consists of the following:

- `bootstrap.home` – The directory in which the Bootstrap code will run.
 - The default location is defined by the system property `user.dir`
 - This can be overridden by a property named *bootstrapHome* in the bootstrap properties file.
- `container.path` – The location of the Container script.
 - The default location is a file named *elf-container.xml* in the current working directory.
 - This can be overridden by a property named *containerPath* in the bootstrap properties file.

Elf Platform Initialization

The `ElfPlatform` handles the instantiation and initialization of the components that are available Elf-wide.

Bootstrap Properties File

A standard properties file.

bootstrap.properties example

```
# ENVIRONMENT CONFIGURATION
containerPath=/Users/arossi/elf-archive/elf-test/elf-container.xml
#bootstrapHome=/tmp/run (defaults to .)

# EVENT MANAGEMENT
eventListener.tag=loggingListener
eventSender.0=tcp://tb1.ncsa.uiuc.edu:61616,ncsa.tools.jms.localevent.JMSLocalEventSender
eventReceiver.0=tcp://tb1.ncsa.uiuc.edu:61616,ncsa.tools.jms.localevent.JMSLocalEventReceiver
eventTopic.0=TROLL

# AUTHENTICATION MANAGER
#authenticationManager.tag=proxyAuthenticationManager
#authenticationManager.proxyAuthenticationManager.username=bjewett

# IDENTITY MANAGER
identityManager.tag=defaultIdentification
identityManager.defaultIdentification.baseId=http://tb1.ncsa.uiuc.edu:8043/broker?
level=DEBUG&groupId=local&node=TEST&user=arossi&workflow=Test
#identityManager.tag=executionServiceIdentityManager
#identityManager.executionServiceIdentityManager.baseId=http://tb1.ncsa.uiuc.edu:8043/broker?
level=DEBUG&groupId=local&node=TEST&user=arossi&workflow=Test
```

Running Elf

Elf requires both `JAVA_HOME` and `ELF_HOME` environment variables to be set. Elf looks for a container in the current directory; this can be overridden by a command-line argument

- `-containerPath <path>`

or a property in `bootstrapProperties` (*containerPath*).

Elf also looks for *bootstrapProperties* in the current directory; this can be overridden by a command-line argument

- `-bootstrapProperties <path>`.

Thus:

- `ELF_HOME/elf`
OR
- `ELF_HOME/elf -containerPath /tmp/foo.xml`
OR

- ELF_HOME/elf -bootstrapProperties /tmp/bootstrap.properties
OR
ELF_HOME/elf -bootstrapProperties /tmp/bootstrap.properties -containerPath /tmp/foo.xml

ELF Container Script Example

```
<elf>
  <configuration>
    <property xmlns:ncsa.updateable.id="submissionType" name="submissionType">
      <value xmlns:ncsa.updateable.id="value">interactive</value>
    </property>
    <property xmlns:ncsa.updateable.id="PWD" name="PWD" type="int" category="workdirEnvironment">
      <value xmlns:ncsa.updateable.id="value">4</value>
    </property>
    <property xmlns:ncsa.updateable.id="TG_CLUSTER_SCRATCH" name="TG_CLUSTER_SCRATCH" type="int" category="workdirEnvironment">
      <value xmlns:ncsa.updateable.id="value">1</value>
    </property>
    <property xmlns:ncsa.updateable.id="SCR" name="SCR" type="int" category="workdirEnvironment">
      <value xmlns:ncsa.updateable.id="value">2</value>
    </property>
    <property xmlns:ncsa.updateable.id="scr" name="scr" type="int" category="workdirEnvironment">
      <value xmlns:ncsa.updateable.id="value">3</value>
    </property>
    <property xmlns:ncsa.updateable.id="file-transfer-protocol" name="file-transfer-protocol">
      <value xmlns:ncsa.updateable.id="value">gridftp://gridftp-hg.ncsa.teragrid.org:2811</value>
    </property>
    <property xmlns:ncsa.updateable.id="ELF_HOME" name="ELF_HOME" category="environment">
      <value xmlns:ncsa.updateable.id="value">/home/ncsa/arossi/elf-latest</value>
    </property>
    <property xmlns:ncsa.updateable.id="TG_CLUSTER_PFS" name="TG_CLUSTER_PFS" type="int" category="workdirEnvironment">
      <value xmlns:ncsa.updateable.id="value">0</value>
    </property>
    <property xmlns:ncsa.updateable.id="JAVA_HOME" name="JAVA_HOME" category="environment">
      <value xmlns:ncsa.updateable.id="value">/usr/local/sunjava/j2sdk1.4.2_13</value>
    </property>
    <property xmlns:ncsa.updateable.id="USER_HOME" name="USER_HOME" category="environment">
      <value xmlns:ncsa.updateable.id="value">/home/ncsa/arossi</value>
    </property>
    <property xmlns:ncsa.updateable.id="service-job-name" name="service-job-name">
      <value xmlns:ncsa.updateable.id="value">MssGet_6012732530767072524</value>
    </property>
    <property xmlns:ncsa.updateable.id="X509_USER_PROXY" name="X509_USER_PROXY" category="environment">
      <value xmlns:ncsa.updateable.id="value">/home/ncsa/arossi/.proxy/x509up_MssGet_6012732530767072524</value>
    </property>
    <property xmlns:ncsa.updateable.id="HOST_NAME" name="HOST_NAME">
      <value xmlns:ncsa.updateable.id="value">tg-login.test</value>
    </property>
    <property xmlns:ncsa.updateable.id="SoftwareStackServiceUrl" name="SoftwareStackServiceUrl" type="java.lang.String">
      <value xmlns:ncsa.updateable.id="value">http://tb1.ncsa.uiuc.edu:8080/softwarestack/services/SoftwareStackRepository</value>
    </property>
  </configuration>
  <workdir />
  <serial-scripts separate-script-dirs="false">
    <ogrescript name="MY_LEAD_INTEGRATION_EXAMPLE">
      <set-system-property name="trebuchet.dir" value="${runtime.dir}" />
      <copy taskName="BULK_FILE_MOVE" target="mssftp://mss.ncsa.uiuc.edu/u/ac/arossi/NAM">
        <configuration>
          <property name="transferMode" value="gridftp-stream"/>
          <property name="target-active" value="true" type="boolean" />
          <property name="tcpBufferSize" value="2097152" type="int" />
        </configuration>
        <listener xmlns:absoluteTag="transfer-completed-listener" bulk="true" useSrc="false">
          </listener>
        <source base="file:/home/ncsa/arossi/elf-test">

```

```

        <include>*</include>
    </source>
</copy>
<copy taskName="BULK_FILE_MOVE" target="mssftp://mss.ncsa.uiuc.edu/u/ac/arossi/NAM2">
    <configuration>
        <property name="transferMode" value="gridftp-stream" />
        <property name="target-active" value="true" type="boolean" />
        <property name="tcpBufferSize" value="2097152" type="int" />
    </configuration>
    <listener xmlns:absoluteTag="transfer-completed-listener" bulk="true" metadataUpdateOnly="false"
useSrc="true">
    </listener>
    <source base="file:/home/ncsa/arossi/elf-test">
        <include>*</include>
    </source>
</copy>
<ftp-chmod recursive="true" permissions="755" dir="mssftp://mss.ncsa.uiuc.edu/u/ac/arossi/NAM2" />
</ogrescript>
</serial-scripts>
</elf>

```