Overview of DAITSS software components

A DAITSS repository consists of five main components that together operate and monitor the repository:

- web services that include User Interfaces,
- daemons that run in the background and manage processing queues and report delivery,
- the command-line submit program that enables DAITSS Operators to submit batches of SIPs to the repository,
- fixity checking scripts that verify the integrity and fixity of AIPs in DAITSS archival storage, and
- DAITSS preservation and storage databases that maintain records of DAITSS repository activity.

DAITSS Web Services

DAITSS has six web services:

1. Core
2. Description Service
3. Action Plan Service
4. Transformation Service
5. XML Resolution Service
6. Storage Master Service
How to Start/Stop DAITSS Web Services

The “daitss” program is used to start and stop all DAITSS Web Services:

- From the system prompt ($) on your DAITSS server issue the following command to start (or stop) all DAITSS web services and daemons:

  ```
  $ sudo /etc/init.d/daitss start
  $ sudo /etc/init.d/daitss stop
  ```

- You will be prompted for your unix user password:

  ```
  $ [sudo] password for <USERNAME>:
  ```

- After you enter your password, the daitss program will display messages about the actions performed.

### DAITSS Daemons

#### Pulse

The Pulse daemon is a program that performs the following tasks:

1. Pulse starts IDLE WIPs/jobs in the DAITSS workspace
2. Pulse evaluates enqueued DAITSS Dissemination and Withdrawal requests and creates WIPs/jobs in the DAITSS workspace when appropriate.

### How to Start/Stop Pulse

The “daitss” command can be used to start and stop the Pulse daemon as follows:
mailer-daemon

The DAITSS mailer-daemon is a program that generates Ingest and Reject reports and either mails them or saves them to disk, depending on system configuration.

The DAITSS mailer-daemon is started and stopped via the daitss start and stop commands used to start all DAITSS services.

DAITSS command-line SIP submission

The DAITSS Workflow Interface provides functionality for the submission of one SIP (Submission Information Package) at a time (see Chapter 6), but often a DAITSS Operator will want to submit batches of SIPs in one operation. This can be done using the DAITSS command-line submission program on the DAITSS server.

Syntax:

submit [options]

The options or flags that can be used with the "submit" command:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>--path</td>
<td>Path on filesystem containing SIPs to submit, either --path or --package</td>
<td>required</td>
</tr>
<tr>
<td>--package</td>
<td>Path on filesystem to a single sip, either --path or --package</td>
<td>required</td>
</tr>
<tr>
<td>--username</td>
<td>Operations agent username</td>
<td>required</td>
</tr>
<tr>
<td>--password</td>
<td>Operations agent password</td>
<td>required</td>
</tr>
<tr>
<td>--note</td>
<td>Submission event note (should be in quotes)</td>
<td>optional</td>
</tr>
</tbody>
</table>
Chapter 3: DAITSS Software Components

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>--source</td>
<td>source of package, which will be used</td>
<td></td>
</tr>
<tr>
<td>--batch</td>
<td>Batch to submit package under</td>
<td></td>
</tr>
<tr>
<td>-d, --delete-orig</td>
<td>Deletes all successfully submitted SIPs from the original location</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

User “jsmith” submits a batch of SIPs located in the /var/daitss/incoming directory, adding a submission note and assigning the packages to a batch named “newspapers”:

```
submit --username jsmith --password ***** --path /var/daitss/incoming --note “UF College of Journalism newspaper” --batch newspapers --source UF &
```

The output log will be sent to the /var/log/daitss/submit-direct directory, and it will be named as follows SUBMIT_<SOURCE>_<YYYYMMHHMMSS>_<BATCH>.log:

```
SUBMIT_UF_201109154517_newspapers.log
```

**Sample log contents:**

```
2011-09-29 15:02:02 ERROR /var/daitss/incoming/bianchi.pdf doesn't appear to be a SIP, skipping

2011-09-29 15:02:14 INFO Thu Sep 29 15:02:14 -0400 2011 --
FDA00067232_00001 -- submitted successfully: EZV5EAWJF_52BB9F

2011-09-29 15:02:47 INFO Thu Sep 29 15:02:47 -0400 2011 --
FDA00028308_00127 -- submitted successfully: E7V1AHHF7_4OUK3K

FDA00028308_00128 -- submitted successfully: ERFV1I0A2_BK8QL6

2011-09-29 15:03:45 INFO Thu Sep 29 15:03:45 -0400 2011 --
FDA00028419_02638 -- submitted successfully: EXXV01XUG_3RMGVV

2011-09-29 15:03:55 INFO Thu Sep 29 15:03:55 -0400 2011 --
FDA01001605_00007 -- rejected: EXJBKV1C7_AFFF3B
```
Chapter 3: DAITSS Software Components

DAITSS fixity checking scripts

DAITSS regularly performs fixity checks of AIPs stored in its repository storage silos to confirm that all AIPs in the DAITSS database exist in storage silos and have not been modified (outside of DAITSS) since the last fixity check. Fixity checking and the scripts used to do this are covered in Chapter 8: Fixity.

DAITSS databases

DAITSS uses multiple PostgreSQL databases:

- **daitss_db**, which contains two schemas:
  - **daitss2**, the DAITSS preservation database
  - **storemaster**, the DAITSS storage database
- **silo_db**, which is a database unique to each DAITSS silo pool

For more information about DAITSS databases, please see the DAITSS Data Entity Dictionary.