

RAPTURE BULK FILE LOADER

september 7, 2012



R E S E A R C H B Y
INCAPTURE

CONTENTS

| | | |
|---|-------------------------------|----|
| 1 | overview | 9 |
| 2 | file name map | 11 |
| 3 | reflex script | 13 |
| 4 | bulk file loader command line | 15 |

LIST OF FIGURES

Figure 1 Bulk File Loader flow 10

LIST OF TABLES

| | | |
|---------|--|--------------------|
| Table 1 | Configuration values in map definition | 11 |
| Table 2 | Bulk File Loader Command Line Parameters | 15 |

OVERVIEW

The Bulk File Loader in **Rapture** is an external application that monitors a well defined folder structure. The names of files that appear in an incoming folder are checked against a set of valid processor configurations until a match is found. That match defines a **Reflex** script that will be invoked for that file. Normally that **Reflex** script will do whatever is necessary to import that type of file and if successful the file will be moved into a done folder. Files that cannot be handled or matched are moved into an error folder.

A typical use of the Bulk File Loader is to sit the application on a remote server, perhaps pointing at an incoming folder that matches a location used by an upload program or an ftp server's home directory. Privileged remote users can upload files to be imported into a **Rapture** system and the behaviour followed for different types of files with different contents can be customized through **Reflex** scripts.

The general flow of the application is shown in [Figure 1 on the following page](#).

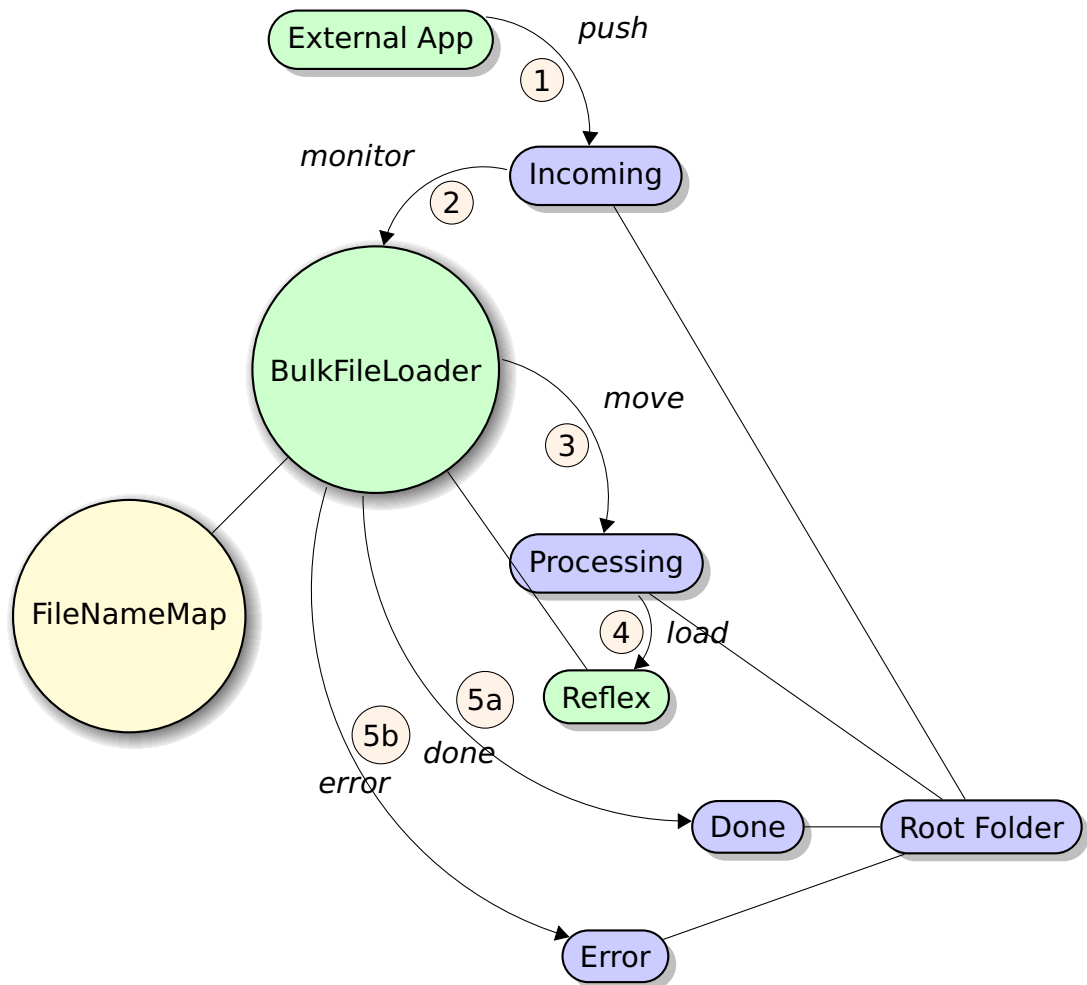


Figure 1: Bulk File Loader flow

In Figure 1 the numbers refer to the following steps:

1. An external application pushes a file to the incoming folder.
2. The Bulk File Loader application is monitoring this folder to detect new files.
3. When it sees a new file it moves the file to the processing folder and resolves using the filename to an appropriate **Reflex** script to process the file.
4. That Reflex script loads the file and does whatever it needs to do with the contents.
5. If successful (a) the Bulk File Loader moves the file into the done folder. If not successful (b) the application moves the file into the error folder.

FILE NAME MAP

The Bulk File Loader is configured through its file name map document – a document stored in **Rapture** and referred to on the command line that invoked the application. The format of this configuration file is reproduced below:

```
{
  "maps" : [
    {
      "nameRegex" : "([^\s]+(\\.(?i)(ord)))$",
      "partition" : "test",
      "scriptName" : "config/orderLoader",
      "params" : {
        "param1" : 1
      }
    }
  ]
}
```

The configuration is basically an array of "maps". Each entry contains the values described in Table 1.

| Field Name | Description |
|------------|---|
| nameRegex | This is a regular expression that needs to match the name of the file for this map to be matched. The example above matches files ending in .ord. |
| partition | The partition in which the script to run for this match is stored. |
| scriptName | The name of the script (in Rapture , in the partition above) that is to be executed for this match. |
| params | Any optional parameters that will be passed to the script on invocation. |

Table 1: Configuration values in map definition

REFLEX SCRIPT

The **Reflex** script that is invoked for a map match should return either true for a successful load or false for an unsuccessful load. All other values (including no return) will be treated as an error by the Bulk File Loader.

The **Reflex** script will be called with the variables pre-initialized to the set of values in the params field of the map that matched (e.g. param1 will have the value 1 from the map definition in the previous chapter) and the predefined variable fileName which will be set to the full path of the file being loaded. This latter parameter will usually be passed into a file **Reflex** function call to setup a file object for processing. A simplistic example is shown in the listing below.

```
1 // A Reflex script to load a file via the bulk file loader
2
3 if !defined(fileName) do
4     throw "Must be called from Bulk File Loader";
5 end
6
7 f = file(fileName);
8 for line in f do
9     // Do something with this line
10 end
11
12 return true;
```

BULK FILE LOADER COMMAND LINE

The parameters for the command line for the Bulk File Loader application are shown in Table 2.

| Name | Example | Description |
|------------|---------------------|--|
| -r | http://rapture | The Rapture environment to connect to |
| -u | bfluser | The user (usually an api key) for the application |
| -p | **** | A password to login to Rapture (if required, not required if the user is an api key user) |
| -f | /user/bfl/data | The root folder to point this loader instance at |
| -partition | config | The partition that holds the configuration |
| -m | config/bfl/orderMap | The configuration path for the mapping document |
| -v | [no value] | Whether the application should present a UI showing the log output |

Table 2: Bulk File Loader Command Line Parameters

A typical invocation of the application would be:

```
java BulkFileLoader.jar -r http://rapture
-u aaadd32342376dafc3334 -f /usr/bfl/data
-partition config -m config/bfl/orderMap
```

The application runs until it has been terminated.

DISCLAIMER

Copyright: Unless otherwise noted, text, images and layout of this publication are the exclusive property of Incapture Technologies LLC and/or its related, affiliated and subsidiary companies and may not be copied or distributed, in whole or in part, without the express written consent of Incapture Technologies LLC or its related and affiliated companies.

©2012 Incapture Technologies LLC

INCAPTURE TECHNOLOGIES LLC
183 Madison Avenue, Suite 801
New York, NY 10016

R E S E A R C H B Y
INCAPTURE