

# Adobe® Device Central: SendToWeb Sample Task

This document describes the `SendToWeb` sample task, an example of a web service plug-in. This plug-in demonstrates how to create an export task for your project that publishes your assets to a web application, in this case an online catalog.

## Features

The `sendToWeb` task demonstrates how to:

- Upload assets through an HTTP connection.
- Verify the response sent by the web service.
- Parse XML responses to show validation or application errors.
- Prompt the user for input when the task has started running.

## Running the Sample

To run this plug-in, you must have a web service available. The Adobe Device Central SDK provides a simple sample application you can run on your local machine, allowing you to run the plug-in against it.

If you want to use your own web service, you can customize the sample plug-in to fit the requirements of your web application.

The following sections describe how to use the web service plug-in.

## Installing the plug-in

Copy the `Sample Plugins/SendToWeb` directory to the user tasks installation directory. For the specific path, see “User tasks folder” in the *Adobe Device Central CS4 SDK Programmer’s Guide*.

## Installing the web service

This SDK provides a Ruby on Rails (<http://www.rubyonrails.org/>) sample web service. To run this application:

1. Install a ruby 1.8.6+ interpreter on your system. Platform-specific instructions are available at <http://www.ruby-lang.org/en/downloads/>.
2. Ensure that you have Rails 2 installed on your server system (at the time of writing, the current version was 2.0.2). Run the following command in your terminal:  

```
gem install rails -v=2.0.2
```
3. The sample web service needs a database to store references to the files uploaded, so ensure that you have SQLite3 installed on your system (<http://www.sqlite.org>).

You can download the precompiled binaries for your platform from <http://www.sqlite.org/download.html>.

- **On Mac OS:** Both Tiger and Leopard come with SQLite3 already installed.
- **In Windows:** Download `sqlite-3_6_0.zip` and `sqlitedll-3_6_0.zip` from the “Precompiled binaries For Windows” section of the SQLite download page. Unzip both files and copy their contents (`sqlite3.exe`, `sqlite3.dll` and `sqlite3.def`) to the `bin` directory within your Ruby installation folder (normally `C:\ruby\bin`).

**NOTE:** The versions of the files mentioned are current at the time of writing.

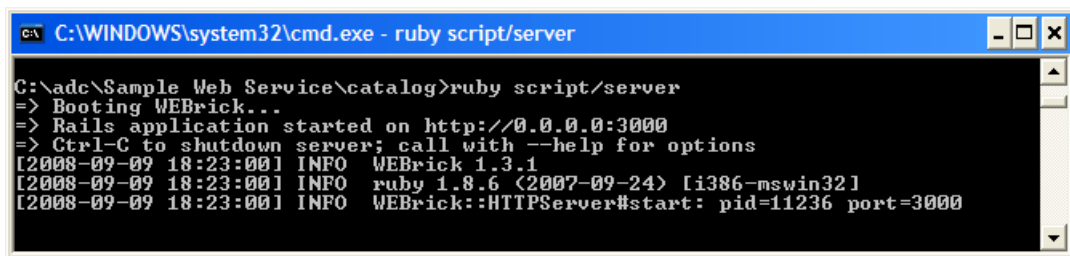
4. Run the following command to enable the web service to use the SQLite3 database:

```
gem install sqlite3-ruby
```

5. Switch to the `Sample Web Service/catalog` folder using your terminal and run the following command to start the web server:

```
ruby script/server
```

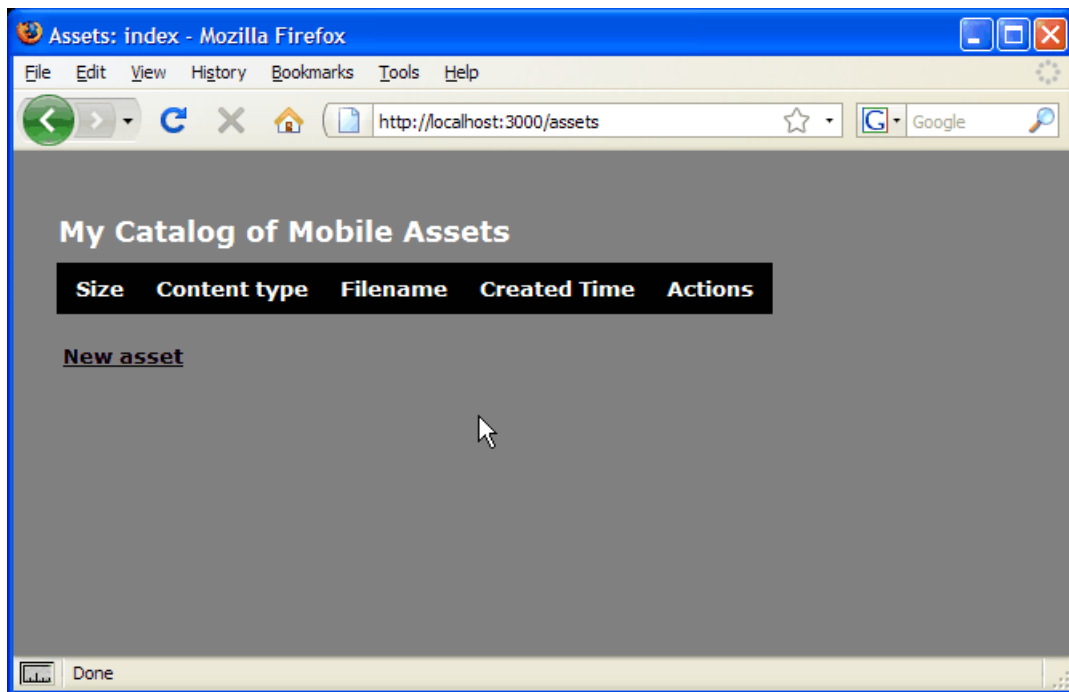
Start-up messages appear, indicating that the application has started.



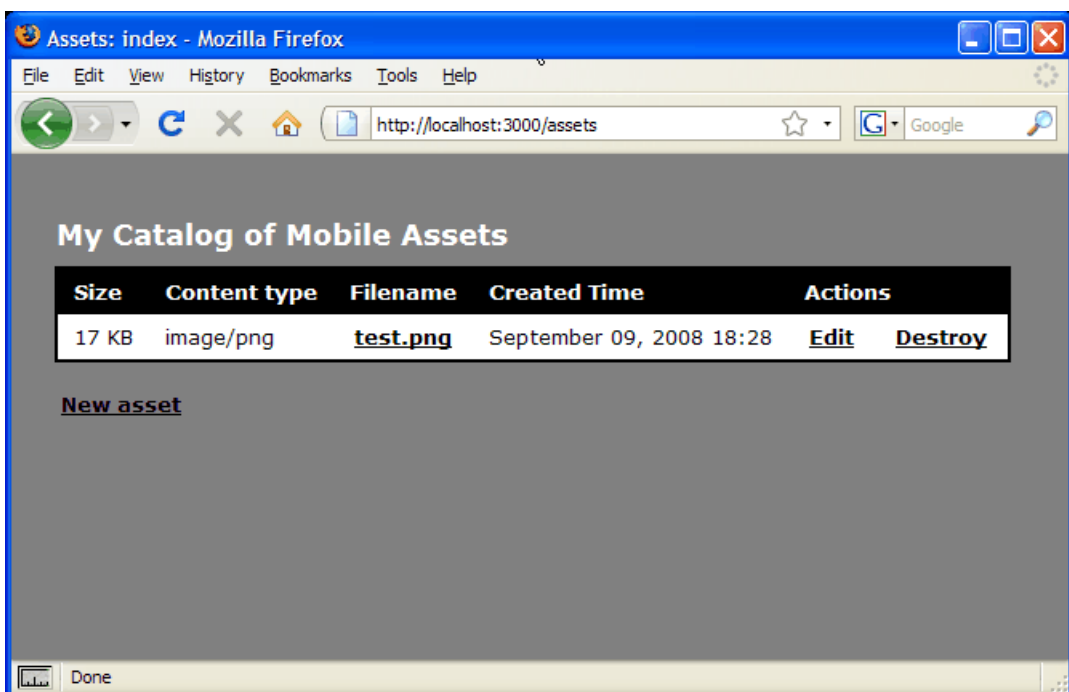
```
C:\WINDOWS\system32\cmd.exe - ruby script/server
C:\adc\Sample Web Service\catalog>ruby script/server
=> Booting WEBrick...
=> Rails application started on http://0.0.0.0:3000
=> Ctrl-C to shutdown server; call with --help for options
[2008-09-09 18:23:00] INFO  WEBrick 1.3.1
[2008-09-09 18:23:00] INFO  ruby 1.8.6 (2007-09-24) [i386-mswin32]
[2008-09-09 18:23:00] INFO  WEBrick::HTTPServer#start: pid=11236 port=3000
```

**NOTE:** Typing `Ctrl+C` in the console running the web server stops the server.

6. From a web browser, go to <http://localhost:3000>.



7. Create a new asset to verify that the setup was successful.



This web service is set up with the following constraints so authentication and application validation can be tested in the plug-in:

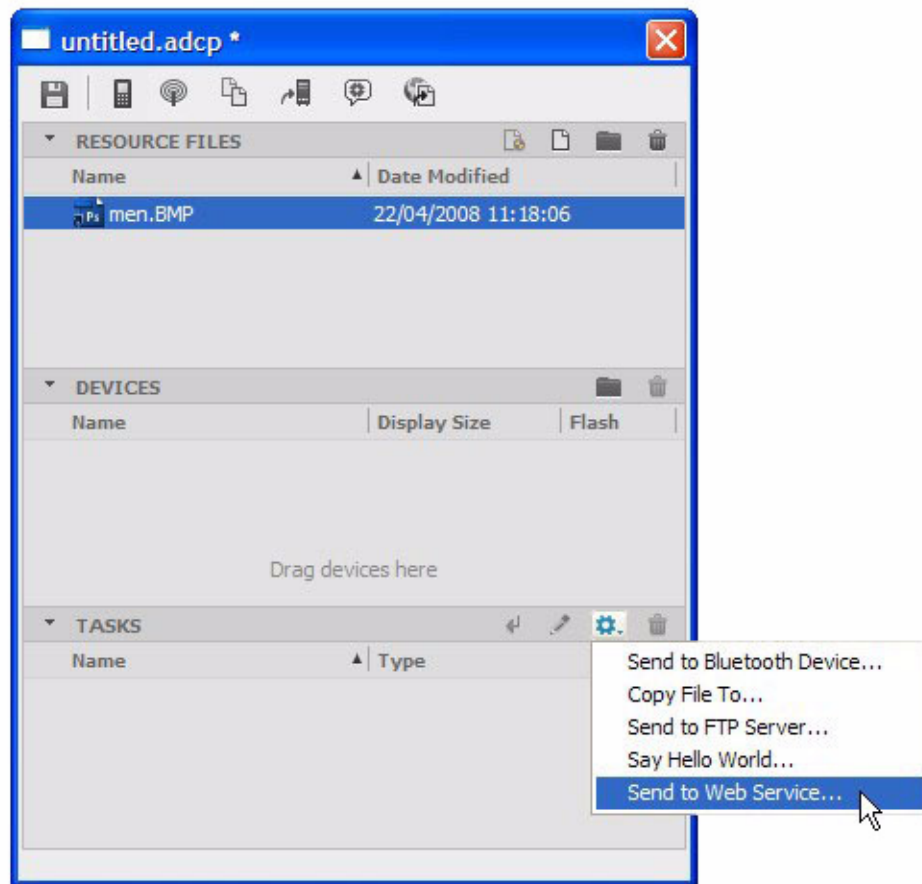
- Uses basic authentication to create new assets:

Username: adc2  
Password: secret

- Assets requirements:
  - Formats supported: GIF, JPEG, PNG, BMP, and SWF
  - Size restrictions: Up to 1 MB per asset

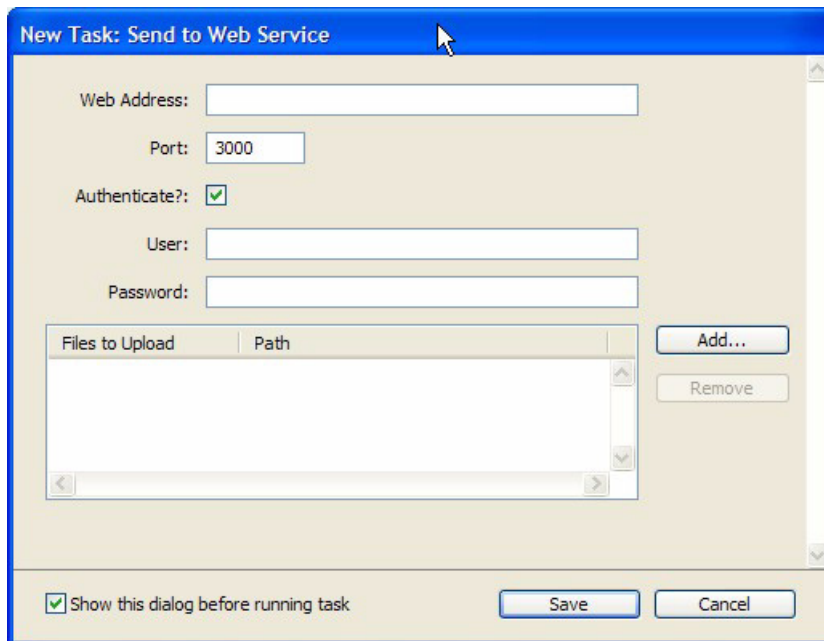
## Creating a new task

1. Open a new project and from **Tasks** menu, create a new Send to Web Service task.

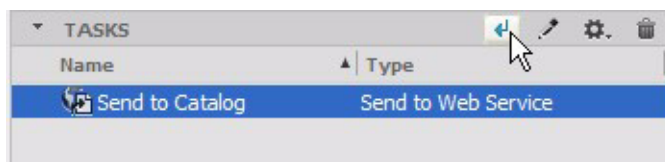


2. Configure the task with the following details:

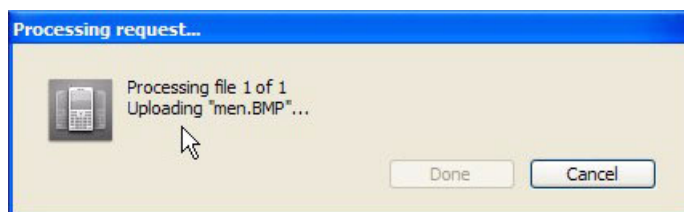
**Web Address:** http://localhost/assets.xml  
**Port:** 3000  
**Authenticate?:** true  
**User:** adc2  
**Password:** secret



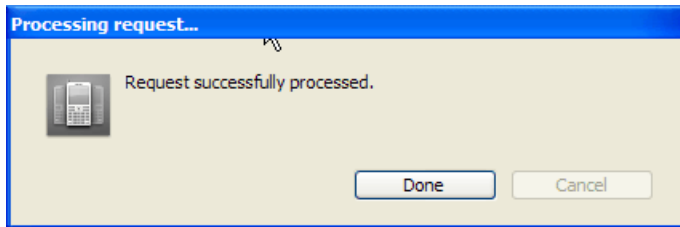
3. Click the **Add** button to select an asset to upload from your project.
4. Save the task and name it `Send to Catalog`.
5. Click the **Run** button.



6. If the task was configured correctly, an upload status dialog box appears.



7. If the request is successful, a success message appears.



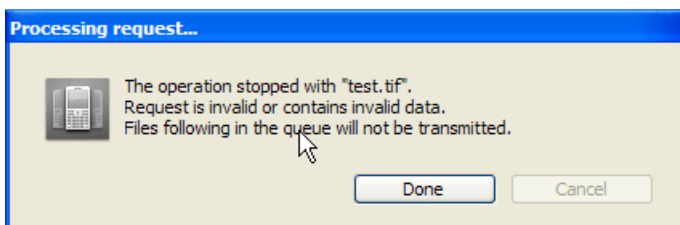
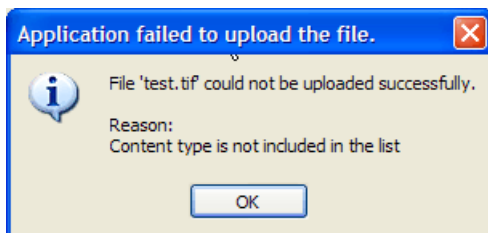
## Testing Error Handling

In testing error handling, you should be familiar with two types of errors:

- [Application validation errors](#)
- [Authentication errors](#)

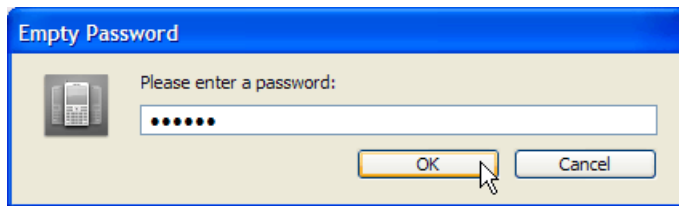
### Application validation errors

If you attempt to upload a file with a format the service does not support, or a file larger than 1 MB, an application error appears, indicating the reason for upload failure. This message is the result of parsing the XML response sent by the web service.

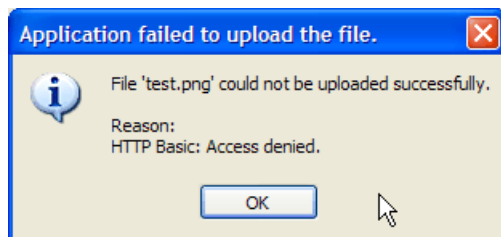


## Authentication errors

During configuration, if you fail to provide a username or password to authenticate the request to the web service, the plug-in gives you another chance to input those values when the task has started running.




If you still fail to enter the correct authentication details, the application reports an authentication error.



## Files

The SendToWeb plug-in consists of the following files:

- WebUpload.xml configuration file
- Web.png icon file 
- WebUpload.jsx ExtendScript file
- HTTPClient.jsx ExtendScript file

These files are part of the Device Central SDK.