PAGER – Publication to ArcGIS Environments & RAMP

Last Updated: June 15th 2015

Contents

[Background 1](#_Toc422134594)

[Prerequisites 1](#_Toc422134595)

[Components 2](#_Toc422134596)

[FileSystemWatcher 2](#_Toc422134597)

[HashCheck 2](#_Toc422134598)

[PAGER\_Scripts 2](#_Toc422134599)

[File System Layout 2](#_Toc422134600)

[Configuration 3](#_Toc422134601)

[filesystemwatcher\_config.json 3](#_Toc422134602)

[PAGER.bat 4](#_Toc422134603)

[The “Payload” 4](#_Toc422134604)

[Process 5](#_Toc422134605)

[Sample PAGER log 7](#_Toc422134606)

[Data Flow Diagram 9](#_Toc422134607)

[Past and Future History of Development 10](#_Toc422134608)

# Background

PAGER – Short for Publication to ArcGIS Environments & RAMP is a combination of a .NET launcher application and Python scripts utilizing Esri’s ArcPy libraries for automating ArcGIS Toolbox related tasks. In its first iteration, PAGER’s focus is to take a geospatial data layer in Shapefile format and transform it into an ESRI REST service with WMS, WFS, KML, GeoJSON, and CSV endpoints. PAGER’s primary focus is to sit between Environment Canada’s Data Catalogue and an ArcGIS Server environment, but its functionality can potentially be extended for automated publication of geospatial data without a Data Catalogue hook.

# Prerequisites

* ArcGIS Server 10.1 – 10.2.2.
* [GeoJSONSOE](https://github.com/geobabbler/AGSOpenFormats) – Server Object Extension for ArcGIS that enabled “Open Data Formats” in ArcGIS service capabilities, permitting distribution via GeoJSON and CSV.
* Python 2.7 or higher.
* Windows Server 2003 or higher with .NET 4.0 or higher.

# Components

## FileSystemWatcher

C# .NET console application. FileSystemWatcher (abbreviated to FSW) is the application launched by PAGER.bat. It continuously scans the specified folder for new zip files dropped off from the Data Catalogue (managed by IMD). When a new zip file is dropped off, FSW extracts the contents, does performs some basic validation, and then launches the ArcPy python scripts to publish the data as a geospatial REST service.

## HashCheck

C# .NET application for verifying file integrity of transferred files. Disabled by default.

## PAGER\_Scripts

These are the python scripts that perform the publication of geospatial data services and the updating of Data Catalogue onlineResources for the supporting metadata to reflect the newly published services.

# File System Layout

* \PAGER\
  + \PAGER\FSW\
    - FileSystemWatcher.vshost.exe.manifest
    - ICSharpCode.SharpZipLib.dll
    - Newtonsoft.Json.dll
    - Newtonsoft.Json.pdb
    - Newtonsoft.Json.xml
    - FileSystemWatcher.exe
    - FileSystemWatcher.pdb
    - FileSystemWatcher.vshost.exe
  + \PAGER\HashCheck\
    - HashCheck.exe
    - HashCheck.pdb
    - HashCheck.vshost.exe
    - HashCheck.vshost.exe.manifest
  + \PAGER\PAGER\_Scripts\
    - editService.py
    - make\_ags.py
    - onlineResources.py
    - onlineResources-manual.py
    - PointColours.lyr
    - PolygonColours.lyr
    - PolylineColours.lyr
    - publisher.ags
    - publisher\_dev.ags
    - publisher\_edn.ags
    - publishService.py
    - PubTemplate10.mxd
    - Readme.txt
    - sendEmail.py
    - workflow.py
    - checkError.py
    - deleteService.py
  + \PAGER\filesystemwatcher\_config.json
  + \PAGER\PAGER.bat

# Configuration

## filesystemwatcher\_config.json

This is the primary configuration file for PAGER. The FSW application seeks out new files for publication and launches the PAGER Python scripts for processing the geospatial data. An example of the configuration file is as follows:

{

"serverName": "Name or Address for ArcGIS Server",

"port": "80",

"agsUsername": "ArcGIS Account",

"agsPassword": "ArcGIS Password",

"templatePath": "C:/PAGER/PAGER\_Scripts/PubTemplate10.mxd",

"connFilePath": "Compile ArcGIS credentials file using make\_ags.py",

"smallkeyPath": [

"config",

"SmallKey"

],

"pubStatus": [

"config",

"Status"

],

"hashPath": [

"config",

"SHA\_256\_HASH"

],

"checkHash": false,

"folder": "data\_donnees",

"geocatUrl": "http://intranet.ecdmp-dev.cmc.ec.gc.ca/geonetwork/srv/eng/metadata.addmapresources",

"geocatUsername": "admin",

"geocatPassword": "badmd1",

"smtpservername": "sdow04exmail2.ontario.int.ec.gc.ca",

"fromaddress": "PAGER-Do-Not-Reply@ec.gc.ca",

"toaddresses":"Separated by semi-colons",

"metaDataUrl": "http://intranet.ecdmp-dev.cmc.ec.gc.ca/geonetwork/srv/eng/csw?service=CSW&version=2.0.2&request=GetRecordById&outputSchema=csw:IsoRecord&id=",

"webAdaptorName": "URL for web adapter if different from serverName"

}

* *serverName* – The ArcGIS server or ArcGIS web adaptor to publish to
* *port* – The network port for ArcGIS server (80 with web adaptor, otherwise typically 6080)
* *agsUsername* – An ArcGIS Server account with permissions to publish services
* *agsPassword* – The password for the ArcGIS Server account
* *templatePath* – The location on disk for the blank .MXD file that the submitted shapefile will be attached to
* *connFilePath* – The location on disk for the .ags ArcGIS Server connection file
* *smallKeyPath* – The key in the submitted Payload File’s JSON descriptor that references the SmallKey value as an identifier.
* *pubStatus* – The key in the submitted Payload File’s JSON descriptor that references the Publication Status code (1, 2, or 3)
* *hashPath* – The key in the submitted Payload File’s JSON descriptor that references the variable for the hash validation.
* *checkHash* – Set to “true” or “false”, this variable enables or disables SHA-256 hash checking to verify the integrity of transmitted Payload files.
* *folder* – The ArcGIS services folder where outputted REST endpoints are published.
* *geocatUrl* – An account with access to update onlineResources for draft metadata records within the EC Data Catalogue.
* *geocatPassword* – The password for the account noted above.
* *smtpservername* – An SMTP mail server for delivery of email logfiles in the event of publication failure or warnings.
* *fromaddress* – The address that appears in the From field of the above noted emails. Default is BASD-PAGER-Do-Not-Reply@ec.gc.ca.
* *toaddresses* – The address that appears in the To field of the above noted emails.
* *metadataUrl* – The URL to call a Data Catalogue CSW service for directly downloading the related XML metadata for the published dataset.
* *webAdaptorName* – In single server environments, this variable should be identical to “*serverName*” noted above. This variable will differ in a load balanced environment in which the web tier is separated from the GIS tier, i.e. *serverName* is “ECQCJ8YwvASP004.ncr.int.ec.gc.ca” and *webAdaptorName* is “intranet.ec.gc.ca”.

## PAGER.bat

PAGER.bat is the console application launcher that passes a few key variables to the FileSystemWatcher application. An example of its format is as follows:

C:

cd C:\PAGER

C:\PAGER\FSW\FileSystemWatcher.exe "E:\ECDMP\_drop" 2000 "C:\PAGER\PAGER\_Scripts\workflow.py" "C:\Python27\ArcGIS10.1\python.exe" "C:\inetpub\wwwroot\pub\_log.txt"

The first 2 lines simply put the script in the application folder’s context. The third line contains 6 parameters:

1. The location of the FileSystemWatcher executable file.
2. The “drop” folder location where files are transmitted to the PAGER application for processing.
3. The interval in which to check for new files, in milliseconds.
4. The location of the parent ArcPy PAGER script.
5. The location of Python’s binary executable.
6. The location to output the logfile.

# The “Payload”

A Shapefile will need to be deposited to the ArcGIS server in a Staging folder. The Shapefile may consist of the following files (from [Wikipedia](http://en.wikipedia.org/wiki/Shapefile)):

Mandatory files:

.shp — shape format; the feature geometry itself

.shx — shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly

.dbf — attribute format; columnar attributes for each shape, in dBase IV format

.prj — projection format; the coordinate system and projection information, a plain text file describing the projection using well-known text format

Optional files:

.sbn and .sbx — a spatial index of the features

.fbn and .fbx — a spatial index of the features for shapefiles that are read-only

.ain and .aih — an attribute index of the active fields in a table

.ixs — a geocoding index for read-write shapefiles

.mxs — a geocoding index for read-write shapefiles (ODB format)

.atx — an attribute index for the .dbf file in the form of shapefile.columnname.atx (ArcGIS 8 and later)

.shp.xml — geospatial metadata in XML format, such as ISO 19115 or other XML schema

.cpg — used to specify the code page (only for .dbf) for identifying the character encoding to be used

The files should be deposited as a compressed ZIP file named after the SmallKey value (i.e. *dd12m.zip*).

Alongside this ZIP file should be a simple json file, named after the SmallKey value as well (i.e. *dd12m.txt*)

This file should contain the fields above, one per line, with the key separated with an = sign, then the value.

For example, dd12m.json will appear as follows:

{

"dataset": {

"UUID": "ada015be-5e32-4815-9607-3edfb9f674bf",

"SmallKey": "dd12m",

"Data\_Type": "",

"Target": "Internal",

"Status": "New",

"From\_Path": "\\EC\_DATA\\001-farming\\CA-SK\\SaskPastures\\AGRICULTURE\_pasture\_boundary\_2009",

"Display\_Field\_En": "StationName\_E",

"Display\_Field\_Fr": "StationName\_F",

"MD5\_Hash": "098f6bcd4621d373cade4e832627b4f6"

}

}

# Process

The PAGER workflow is a component of a larger publication workflow initiated by the EC Data Catalogue. This process is detailed as follows, with highlighted sections relevant to PAGER/RCS/RAMP components hosted and supported by BASD.

|  |  |  |
| --- | --- | --- |
| **Step** | **Task/Job** | **Actor/Component** |
| 1 | Logs in as Editor | Data Catalogue |
| 2 | Creates New Metadata record | Data Catalogue |
| 3 | Creates New Collection Metadata | Data Catalogue |
| 4 | Creates New Product Metadata | Data Catalogue |
| 5 | Uploads Geospatial Zipped SHP file | Data Catalogue |
| 6 | Checks if GDP is type=shape | DC Temporary Data Folder |
| 7 | Validates GDP using GeoTools | DC Temporary Data Folder |
| 8 | Presents Display Field Selection to | ODS |
| 9 | Saves GDASP.zip to | Data Catalogue Staging Area |
| 10 | Saves and Exits MD record | Data Catalogue |
| 11 | Tells the file(s) and path(s) that need convertion in the | Data Conversion Utility |
| 12 | Unzips GDP | Data Conversion Utility Temporary Folder |
| 13 | Converts GDP to open formats | Data Catalogue Staging Area |
| 14 | Confirms Data conversion Started to | ODS |
| 15 | Converts dataset to GeoJSON | Data Catalogue Staging Area |
| 16 | Converts dataset to CSV | Data Catalogue Staging Area |
| 17 | Updates online resources field in Metadata record with converted fileds in | Data Catalogue |
| 18 | Confirms data is converted | ODS |
| 19 | Submits record for Publishing Approval | Data Catalogue |
| 20 | Adds Submission request to Pending Processing List | Data Catalogue |
| 21 | Creates GDP ZIP package for ESRI server | DC Temporary Data Folder |
| 22 | Creates Small Key (UUID) JSON file | DC Temporary Data Folder |
| 23 | Adds Shape contents to UUID.zip | DC Temporary Data Folder |
| 24 | Adds Metadata.xml (ISO) to UUID.zip | DC Temporary Data Folder |
| 25 | Adds Smallkey.json file to UUID.zip | DC Temporary Data Folder |
| 26 | Creates DataMartPackage.zip (DMP) | DC Temporary Data Folder |
| 27 | Copies DMP.zip to | Data Publication Utility Temporary Folder |
| 28 | Triggers Internal Publication process | Data Publication Utility |
| 29 | Moves DMP.zip from the Data Publication Utility Temporary Folder to | Data Mart |
| 30 | Decompresses the DMP.zip, creates Data Mart Hierarchy and content deployed in the | Data Mart |
| 31 | Updates online resources with new links/location in Data Mart | Data Catalogue |
| 32 | Add View Data Mart Link pointing to root of product | Data Catalogue |
| 33 | Deposits GDP into OUTGOING feed folder | MetPX Outgoing Feed folder |
| 34 | Picks up GDP from Outgoing Feed Folder (5 minutes) | MetPX Service Memory |
| 35 | Delivers GDP to | PAGER SFTP Drop Folder |
| 36 | Monitors PAGER Server Folder | PAGER SFTP Drop Folder |
| 37 | Unzips GDP | PAGER Server |
| 38 | Creates Web Services | PAGER Server |
| 39 | Creates ESRI REST service per product | ArcGIS Server |
| 40 | Creates ESRI WMS per product | ArcGIS Server |
| 41 | Creates ESRI WFS per product | ArcGIS Server |
| 42 | Creates ESRI KML per product | ArcGIS Server |
| 43 | Creates ESRI JSON created per product | ArcGIS Server |
| 44 | Updates MD record with Service URL | Data Catalogue |
| 45 | Adds Service URL record to | RCS |
| 46 | Adds View on Map URL to Metadata record | Data Catalogue |
| 47 | Updates Pending Request List | Data Catalogue |
| 48 | Adds Submission request to Submission Request List | Data Catalogue |
| 49 | Sends Request for Approval E-Mail to | Level 1 - Content Reviewer |
| 50 | Updates Pending Request List | Data Catalogue |
| 51 | Receives e-mail - level 1 and level 2 Publication Request | Data Catalogue |
| 52 | Reviews Pending Request list for level 1 and level 2 approval | Data Catalogue |
| 53 | Approves Publication Request | Data Catalogue |
| 54 | Sends Request for Approval E-Mail to | Level 2 - Content Reviewer |
| 55 | Sends Approval Level-1 Notification email to | Metadata Group Informant |
| 56 | Receives e-mail - level 2 Publication Request | Data Catalogue |
| 57 | Reviews Pending Request list for level 2 approval | Data Catalogue |
| 58 | Approves Publication Request | Data Catalogue |
| 59 | Sends approval level 2 notification mail to | Level 1 - Content Reviewer |
| 60 | Makes Metadata record internally available | Data Catalogue |
| 61 | Appends metadata information for inline display | RCS |
| 62 | Triggers External Publication process | Data Catalogue (External) |
| 63 | Send MEF File to | External Data Catalogue |
| 64 | Confim Published Data Package (Internal and External) | Data Mart |
| 65 | Confirm dataset-shp.zip is present | Data Mart |
| 66 | Confirm datapackage.json is present | Data Mart |
| 67 | Confirm dataset.geojson is present | Data Mart |
| 68 | Confirm dataset.csv is present | Data Mart |
| 69 | Confirm sub directorires and files present | Data Mart |
| 70 | Confim Published Data Services (Internal and External) | Data Catalogue |
| 71 | Confirm Data Available in (Internal) | RAMP Viewer |
| 72 | Confim View on map works | RAMP Viewer |
| 73 | Confirm Add to Map Preview works | RAMP Viewer |
| 74 | Confirm MD added to CSW service | Data Catalogue |
| 75 | Confirm that Metadata can be searched | Data Catalogue |
| 76 | ConfirmMD Record returned in search results | Data Catalogue |
| 77 | ConfirmView MD record | Data Catalogue |
| 78 | Confirm Add dataset to map | Data Catalogue |
| 79 | Access service URL | ArcGIS Server |
| 80 | Access Data | Data Mart |
| 81 | View in Mart | Data Mart |

# Sample PAGER log

Below is an excerpt from PAGER’s console/logfile of a successful publication request from the Data Catalogue:

Press 'q' to quit.

New File Found, But Not a Zip File: fe36798c-e718-447e-a7ad-47cf83a0bf7a.zip.tmp

----------------------------------------------------------------------------

2015-06-15 11:06:18

Payload Found, Unzipping: fe36798c-e718-447e-a7ad-47cf83a0bf7a.zip

Hash check skipped.

Executing script with the parameters:

Smallkey: fe36798c-e718-447e-a7ad-47cf83a0bf7a

Payload Path: E:\ECDMP\_drop\fe36798c-e718-447e-a7ad-47cf83a0bf7a

Server Name: gibson

Port: 80

Template Path: C:/PAGER/PAGER\_Scripts/PubTemplate10.mxd

Connection File Path: C:/PAGER/PAGER\_Scripts/publisher\_edn.ags

Publish Status: 2

Folder: data\_donnees

MakeDescriptor URL (DEPRECATED):

Catalogue URL: http://intranet.ecdmp-dev.cmc.ec.gc.ca/geonetwork/srv/eng/metadata.addmapresources

Catalogue User: admin

ArcGIS User: arcgis

Email Server: sdow04exmail2.ontario.int.ec.gc.ca

Email sender: BASD-PAGER-Do-Not-Reply@ec.gc.ca

Email receiver(s): Dan.Bowerman@ec.gc.ca

Catalogue Metadata URL: http://intranet.ecdmp-dev.cmc.ec.gc.ca/geonetwork/srv/eng/xml.metadata.get?uuid={0}

Web Adaptor Name: 142.135.82.123

Script exited with code: 0

Output:

Start unzipping to folder: E:\ECDMP\_drop\fe36798c-e718-447e-a7ad-47cf83a0bf7a

Validating shapefile...

CAN\_Major\_Watersheds\_NAD83.shp exists

CAN\_Major\_Watersheds\_NAD83.shx exists

CAN\_Major\_Watersheds\_NAD83.dbf exists

CAN\_Major\_Watersheds\_NAD83.prj exists

access metaData, We failed to reach a server.

Reason: Internal Server Error

Attempting to update the service: data\_donnees/fe36798c-e718-447e-a7ad-47cf83a0bf7a

Service does not exist. Publishing as new service.

Creating MXD...

Publishing MXD created

Publishing MXD in: E:\ECDMP\_drop\fe36798c-e718-447e-a7ad-47cf83a0bf7a\fe36798c-e718-447e-a7ad-47cf83a0bf7a

maxRecordCount set to: 10000

maxInstances set to: 1

WMS/WFS service properties set

Service definition created with WMSServer, WFSServer, GeoJSONServer enabled

The following information was returned during analysis of the MXD:

----MESSAGES---

(u' Layer draws at all scale ranges CODE 30003 applies to:CAN\_Major\_Watersheds\_NAD83',)

----WARNINGS---

Map is being published with data copied to the server using data frame full extent CODE 10045 applies to:

(u" Layer's data source is not registered with the server and data will be copied to the server CODE 24011 applies to:CAN\_Major\_Watersheds\_NAD83",)

Missing Tags in Item Description CODE 24059 applies to:

(u" Layer's data source doesn't have a spatial index CODE 10002 applies to:CAN\_Major\_Watersheds\_NAD83",)

Missing Summary in Item Description CODE 24058 applies to:

----ERRORS---

None

Service successfully published

OnlineResources update successful

# Data Flow Diagram



# Past and Future History of Development

Original Development (ECDMP Phase 1) – F.Y. 2013 – Project called “ECDMP Publication Workflow”

* Dan Bowerman (Project Lead)
* Jianfeng Zhao (Esri Canada)
* Shimu Wu (Co-Op Student)

Extended Development (ECDMP Phase 2) – F.Y. 2014 – Project named to “PAGER”.

* Dan Bowerman (Project Lead)
* Hua Wan (Esri Canada)
* David Cacilhas (Co-Op Student)
* Nino Dragic (Co-Op Student)

Version 2.0 – Robustness development and additional formats support – F.Y. 2015

* Dan Bowerman (Project Lead)
* Serguei Zinine (EC Lead Developer)
* Spencer Wahl (EC Co-Op Student)
* Cynthia Li (EC Co-Op Student)