Addendum: Python Sample Scripts

Python Programming Language Examples

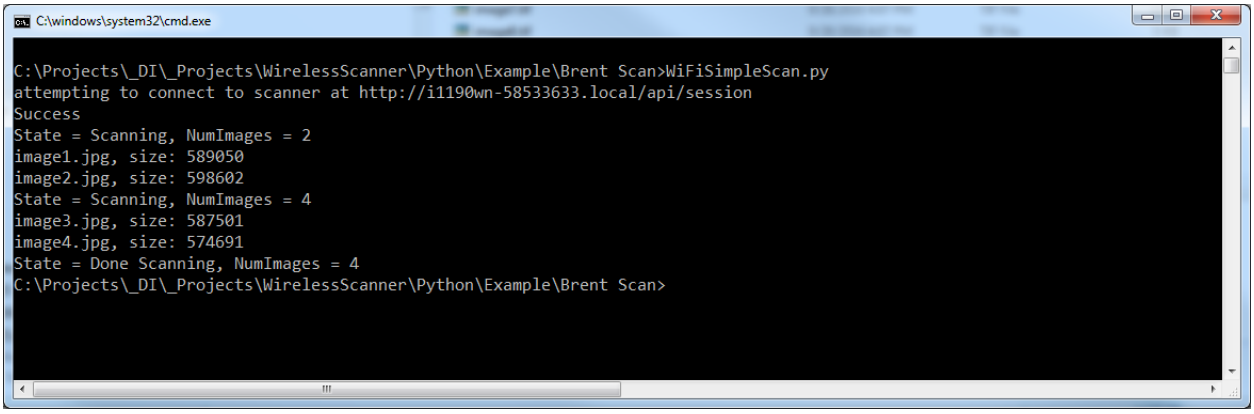
Python is a widely used high-level, interpreted programming language. Python interpreters are available for many operating systems. We provide three examples that demonstrate scanning and use of the session RESTful Web API. Each of these three examples are provided in both source and executable form to enable you to use them as both reference or cut and paste within your python code. They also represent the proper mechanism for driving the scanners Web API.

Example: WifiSimpleScan.py

This is a console application that when run, does a scan and storage of the scanned images into the current folder. This is less than 50 lines of code. You will need to edit the line:
url = "http://i1190wn-58533633.kodakalaris.net/api/session"
to your scanners IP address, for example: http://10.0.0.1/api/session.

This code demonstrates a wait for the scanner to wake, commands to perform a scan of any number of pages, and storage of those images as either TIFF or JPG based on the current configuration of the scanner. The configuration command, PUT /api/session, is not used in this example. Finally, termination (delete) of the scan session.

The example output would look like:

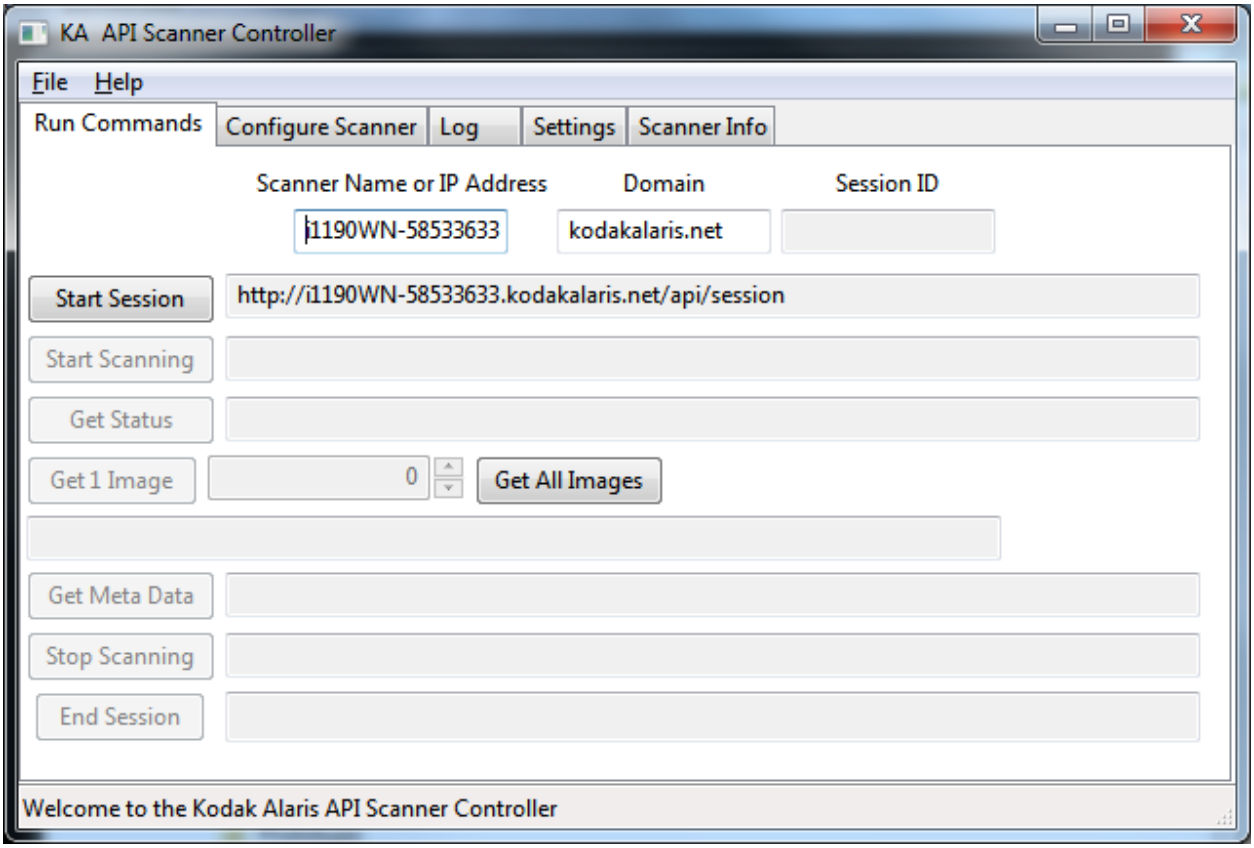


This does not address error handling.

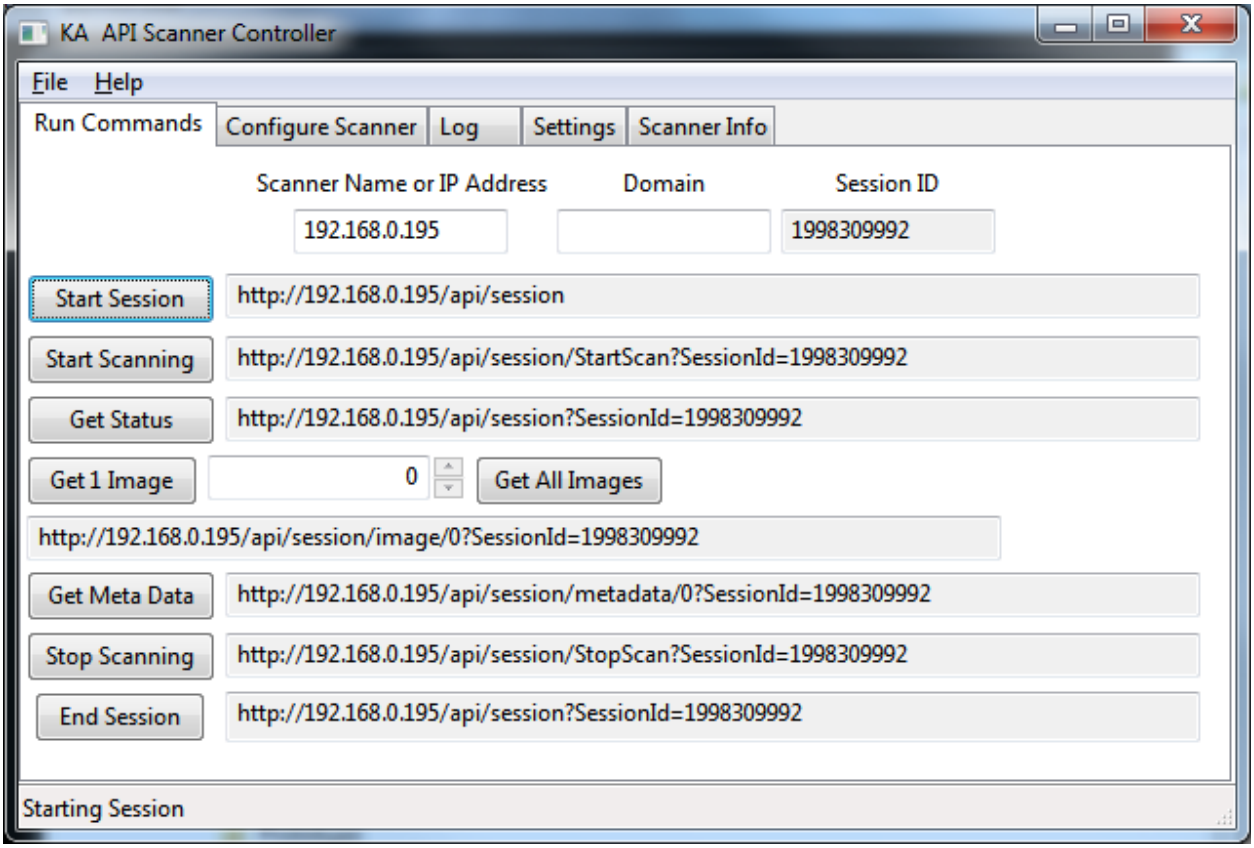
Example: KaApiScannerController

This example is close to 1000 lines of code. It implements a user interface. It is designed to expose the Web API as single button presses and implements the POST, PUT, DELETE, GET that is appropriate for the button. Like the http://<IP>/conf/diag.html page but written in Python.

1) This initial screen requires you to edit the "Scanner Name or IP Address":

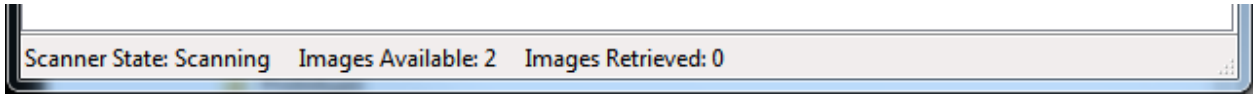


Once you edit the IP and press "Start Session", the scanner shows that it is in use and the screen updates to show:



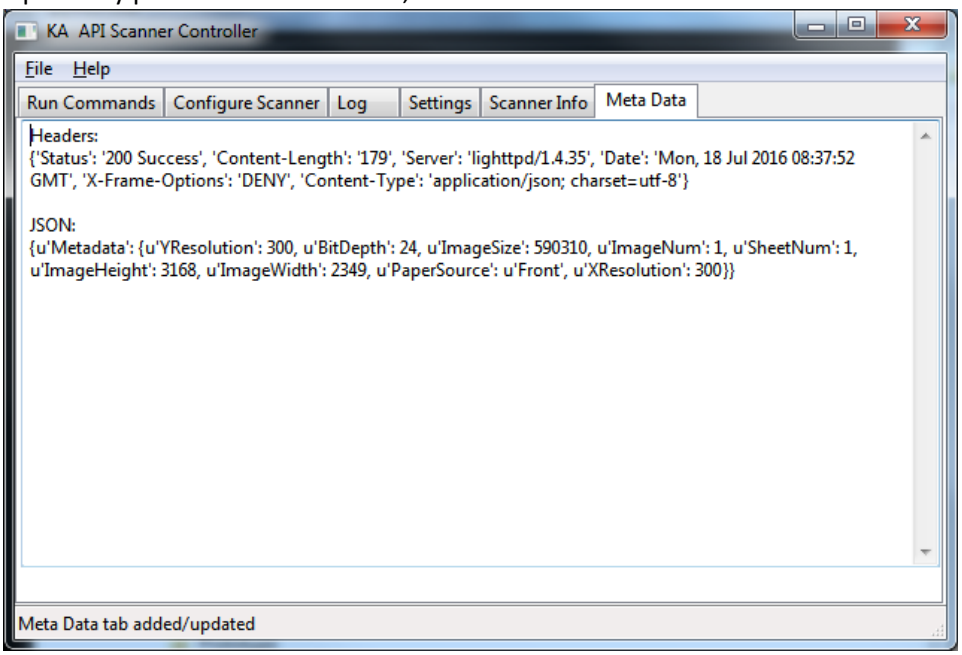
At this point you can complete a scan by:

2) Optionally press the “Configure Scanner” tab, to change the default scanner settings. Within that tab you would press “Get Config”, to populate the scanner settings. Pull down on any of the settings fields to change them, and when done, press Set Config to change the session configuration.
3) Now press the “Run Commands” tab.
4) Press the “Start Scanning” button. Notice the display on the scanner changes.
5) Optionally press “Get Status” to show the current state of the scanner, and number of pages scanned.
6) At this point you would put paper in the scanner and press the START > button.
7) Press “Get Status” while the pages are being scanned to see the Scanner State and Images Available update:



8) Keep pressing “Get Status” until “Images Available” increases. At this point you can press “Get 1 Image” or “Get All Images”. Note that this button also implements a Get Status. The Get Status is important to update the Web API with the updated state prior to reading the images.

9) Optionally press “Get Meta Data”, and review the Meta Data tab.



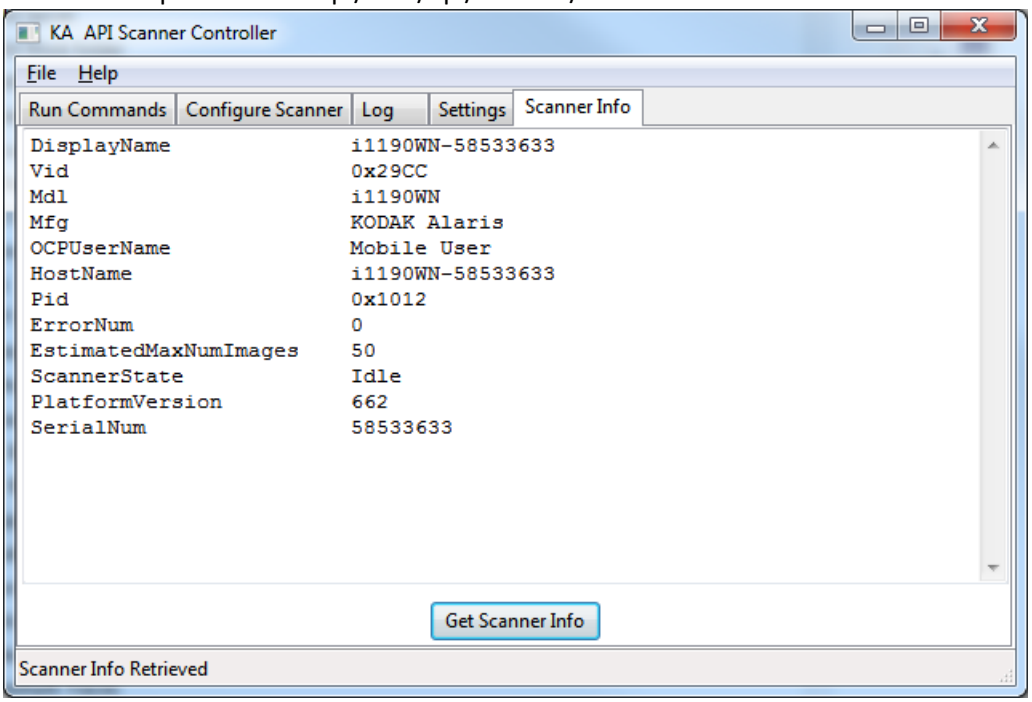
10) Stop Scanning is only used in the middle of a scan job to end the current page being scanned and to stop paper movement. It does not end the session. You should still “Get Status” and wait for the “Scanner State” to go to “Done Scanning” or “Error”. Notice the scanner display when you press Stop Scanning. Also notice that you retain ownership of the scanner once the “Start Session” is successfully obtained. You would now either “Start Scanning” again to reset the job, delete the current images, and start a new scan or you would “End Session”.
11) Press “End Session” to complete this scan session and to delete all internal images from the job. This will also occur after five minutes of inactivity. The “End Session” can be executed any time within a job to abort the current job.

Other tabs in the KaApiScannerController application

The Log tab shows a running log of commands and responses while issuing commands on the various tabs.

The Settings Tab enables you to choose a filename prefix and an output destination for downloaded images.

Scanner Info performs a http:/<IP>/api/scanner/status command:



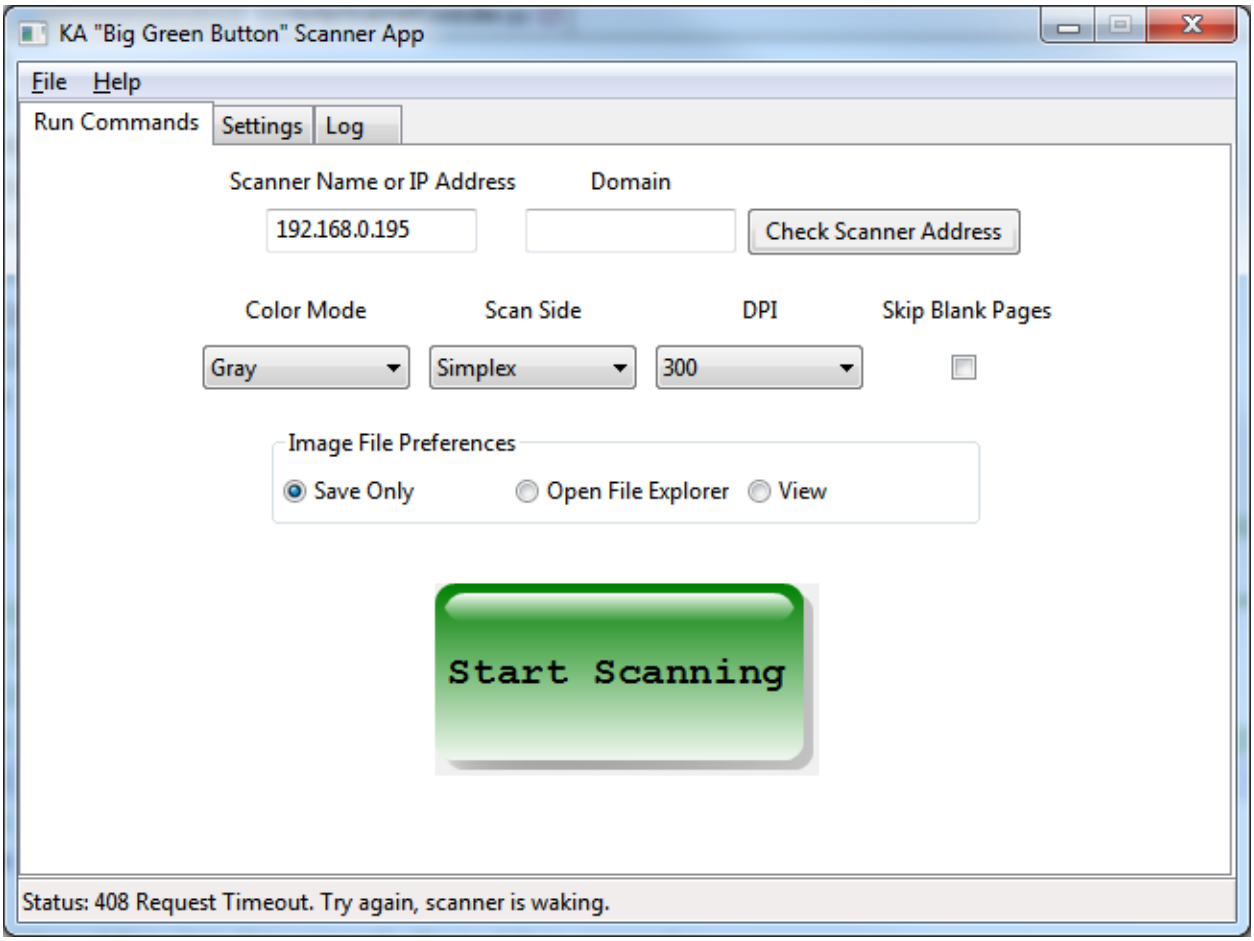
Closing the application will automatically send an “End Session” if one has not already been sent to the scanner.

Example: KaBgbScanningApp.py

Up to this point you've seen an application that does a simple scan, a developer's tool to play around with the API without writing any code, and now we show an application that a product might be designed to provide. The KaBgbScanningApp is the "Big Green Button" application, which shows the configuration settings and a button to perform a scan using those configuration settings.

Put in the IP address of the scanner you are connecting to and press the red "Check Scanner Address" button. This loads the configuration from the scanner and presents it with the list of options available. It offers you an option to save the scanned images, Open once done scanner, or view the scanned images.

Pressing the Big Green button "Start Scanning" demonstrates how an application might choose to use the Web API and the parameter choices to perform a scan.



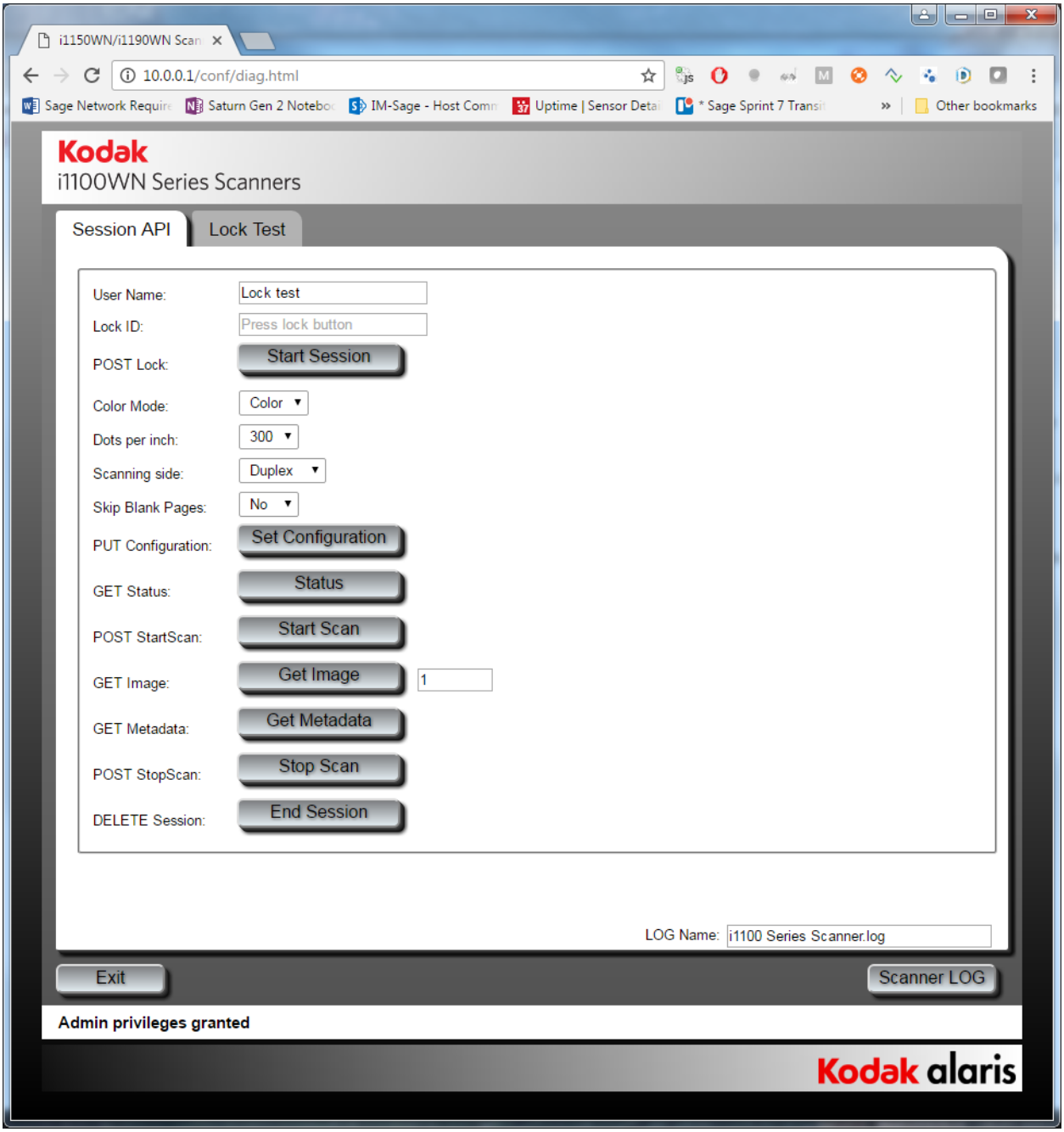
The settings tab enables you to choose a display name which shows up on the scanner display, showing the scanner is in use by "name". The settings tab enables you to choose a "Filename prefix" for the images stored to your disk, and an "Output Destination" for where you'd like to store those images.

The Log tab shows a running log of commands and responses while issuing commands on the various tabs.

Replacement in Starters Guide

The Web API within the scanner can be exercised one API call at a time via a webpage on the scanner:

http://10.0.0.1/conf/diag.html



The first tab shows each of the API calls with a short description to its left, which informs you whether the button does a GET, POST, DELETE, or PUT. Feel free to press F12 to look at the code behind each of these buttons. There is a wealth of information exposed by the web pages and JavaScript within those web pages to drive functionality. Press some buttons and start interacting via our RESTful Web API.