METHOD FOR A GRAPHICAL USER INTERFACE TO AID AN ADMINISTRATOR IN CREATING PROGRAMS THAT INTERACT WITH A DEVICE VIA AN APPLICATION PROGRAMMING INTERFACE

Jay Johnston
Magnus Mortensen
David White

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Johnston, Jay; Mortensen, Magnus; and White, David, "METHOD FOR A GRAPHICAL USER INTERFACE TO AID AN ADMINISTRATOR IN CREATING PROGRAMS THAT INTERACT WITH A DEVICE VIA AN APPLICATION PROGRAMMING INTERFACE", Technical Disclosure Commons, (August 15, 2018) https://www.tdcommons.org/dpubs_series/1413

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
METHOD FOR A GRAPHICAL USER INTERFACE TO AID AN ADMINISTRATOR IN CREATING PROGRAMS THAT INTERACT WITH A DEVICE VIA AN APPLICATION PROGRAMMING INTERFACE

AUTHORS:
Jay Johnston
Magnus Mortensen
David White

ABSTRACT
An Application Programming Interface (API) Workflow Recorder is described herein which enhances the human Graphical User Interface (GUI) of a device to directly expose the programmatic methods that can be used to make changes to the device. An administrator may enable the API Workflow Recorder, make changes to the device using familiar point-and-click workflow in the GUI, and then generate a programmatic script for a machine that uses API calls to accomplish the same tasks that were just performed in the GUI, but using the API for the device. This system greatly improves the speed and accuracy for creating offline programs that interact with a device via the device’s API.

DETAILED DESCRIPTION
An end user’s experience with a device is dependent on developers. In particular, more and more developers are tasked with programming systems that interact with Application Programming Interfaces (APIs), and that development process is unnecessarily hard.

More users are moving to API control for their network devices, yet it is very difficult for programmers to determine the specific device API changes required (and in what order) to accomplish a specific task on a networking device.

API documentation is hard to parse, and while it might reveal how to accomplish one specific action, it does not explain the flow of discrete changes required to accomplish a workflow on a device.

Humans can usually learn in a Graphical User Interface (GUI) the flow of how to make the required changes to their product, since the GUI is presumably intuitive. But the intuitiveness of the GUI is completely lost when the programmer goes to an API explorer to try and accomplish the same actions the user did in the GUI.
What is needed is an intuitive system that aids an administrator in easily creating an offline API program that accomplishes the same workflow (set of changes) that the administrator already knows how to accomplish using the GUI.

Accordingly, described herein is a method that allows humans to leverage their experience and knowledge of a GUI to create API programs. This works by enhancing the human GUI of a device to directly expose the programmatic methods that can be used to make changes to the device.

The machine API methods for making various changes are exposed right alongside the human-facing widgets and controls in the GUI that the human is used to working with. This allows the human to directly map their known and familiar workflows with the machine API changes needed to make similar changes to the device.

The human can learn a workflow in the GUI and then directly translate that into the machine program to make those API changes to accomplish the same goal.

First, the device user interface exposes the programmatic API access for various actions and methods right alongside the GUI widgets for the device. Second, the administrator enables the API Workflow Recorder feature to start the next phase. Third, the administrator begins working in the GUI as they normally would to accomplish a task.

At each step the system records the complementary API actions required to make the same change as the administrator is making in the GUI. When complete, the administrator stops the API Workflow Recorder. The administrator can then use the API Workflow Recorder to compile a real-world working program which includes all the API changes needed to duplicate the changes they just made in the GUI, but using a machine.

The script may then be replayed by a machine, used as a template for future changes, or simply modified to meet the specific need of the developer. This technique allows developers and administrators to rapidly develop the API programming they need to accomplish real-world tasks using programmatic control of a device.

Figure 1 below illustrates enabling the API Workflow Recorder feature.
Figure 1

Figure 2 below illustrates starting the activation and recording of the user’s interaction with the GUI.

Figure 2

Figure 3 illustrates an indication that the API recording has begun.
Figure 3

Figure 4 illustrates recording the API information needed to accomplish a first example task in the workflow.
Figure 5 below illustrates recording the API information needed to accomplish a second example task in the workflow.

![Image of API workflow recorder]

**Figure 5**

Figures 6-9 below illustrate recording the API information needed to accomplish a third example task in the workflow.
Figure 6

Now the user adds a policy referencing the object created earlier.

Figure 7

The API Workflow Recorder now has 2 step recorded. The object creation and the policy configuration.
Figure 8

Figure 9

Figure 10 below illustrates exporting the machine code, with the specific API actions needed, to accomplish this same task.
The resulting machine code may be downloaded, and all steps are available for modification by the programmer if needed. Figure 11 below illustrates the complete generated code.
Figure 11

Figure 12 below illustrates the snippet of the specific APIs at each step that were recorded by the system.
In summary, an API Workflow Recorder is described herein to enhance the human GUI of a device to directly expose the programmatic methods that can be used to make changes to the device. An administrator may enable the API Workflow Recorder, make changes to the device using familiar point-and-click workflow in the GUI, and then generate a programmatic script for a machine that uses API calls to accomplish the same tasks that were just performed in the GUI, but using the API for the device. This system greatly improves the speed and accuracy for creating offline programs that interact with a device via the device’s API.