

HAPS Code README

Updated 2025-05-06

Table of Contents

- [Summary](#)
- [AHK](#)
- [JavaScript](#)
- [HTML](#)

Summary

The HAPS (HOLOGIC Automated Print Screen) tool was designed to take multiple Print Screens of the GUI (running in Demo Mode) in every available language in the GUI System Settings. The tool uses AHK (AutoHotkey) to control the mouse and keyboard to take the Print Screens. There is also a HAPS webpage that offers the user a graphical interface to select the needed screens. This webpage also contains specific notes for some screens that may not be supported, or require additional steps (i.e., a thumb drive inserted for the "Export Designs" button). Currently the HAPS program supports the T5000 Admin Options, Barcode Config, and Label Design screens, although not all screens are supported.

AHK

Current Version: HAPS.ahk :: v1.0a :: 2600 lines :: 2025-05-06

Summary:

HAPS uses AutoHotkey v1.1 (1.1.37.02 - March 16, 2024) to control the mouse and keyboard. Each custom function will navigate through the GUI (Demo Mode) and take a Print Screen. Each custom function will always start and finish on the home screen. After every Print Screen is taken, HAPS will then change the language of the GUI, and then retake each Print Screen. This will repeat for every available language in the GUI. Each Print Screen is saved as a PNG with the filename being the official Screen ID (i.e., AO-000013) followed by the language (i.e., AO-000013_French.png).

Requirements:

- Install AutoHotKey v1.1
- T5K GUI running in Demo Mode
- Default screen resolution (this is discussed in detail below)
- No other tasks running that may manipulate the mouse or keyboard

Screen Resolution:

HAPS is programmed to move the mouse to exact coordinates based off of the GUI running at the native 640x480 resolution. Sometimes when running the GUI on different monitors with different resolutions or zooms, the GUI will run at a resolution that is different from 640x480. There is a CONSTANT in the AHK code to adjust for the difference in resolution (i.e., 1.25 for 125% of native 640x480 resolution). Generally speaking, this resolution is easy to discover, and will be 1.00 (default), 0.50, 1.25, or 1.50. The HAPS Operators Manual covers in detail how to discover the current resolution of the GUI.

Code Architecture:

The code uses a basic loop, with a nested loop, to capture every screen in every language. The outer loop uses the number of languages, while the inner loop uses the array of screen IDs. This means all screens for one language will be completed before changing to the next language. After the inner loop completes, there is a custom function to update the language. On the last iteration, the language is reset to English. The inner loop calls a single function which only looks at the current screen ID, and calls the custom function based on the screen ID. There is a custom function for every screen ID. The custom function simply calls a series of mouse clicks to navigate to the proper screen, then calls a Print Screen, then navigates back to the home screen. Most

of the mouse clicks are for “standard” button locations in the GUI, but there are some custom mouse click coordinates needed. The Print Screen function simply “presses” the Print Screen key, then the GUI Demo Mode will open the Save As window, and the function will type in the current screen ID and language for the file name. Lastly there are two language specific functions; one to update the GUI to the next language, and the other to reset the GUI to English.

Code Outline:

- AHK Initialization
- CONSTANTS
 - LANGUAGE_COUNT
 - RESOLUTION
- Screen ID Array
- Mouse Initialization
- Main Loop
- Screenshot Functions (check current screen ID and call custom function)
 - Admin Options
 - Barcode Config
 - Label Design
- Custom Functions (one for each screen)
 - Admin Options
 - Barcode Config
 - Label Design
- Button and Print Screen Functions
 - “Standard” Buttons
 - Click Function
 - Print Screen (and Save As)
- Language Functions
 - Next Language
 - Reset Language (to English)

Updating Code to Add Screens:

- Add the screen ID and custom function call to the Screenshot Functions list (in the appropriate category, i.e., Barcode Config)
- Add the custom function (in the appropriate category)
- Use any “standard” buttons
- Can call ClickCustom(x, y) directly with exact coordinates
- Can create a new “standard” button in the ClickButton(b) function (in the appropriate category)

Special Functions:

- There are certain screens that can be captured using custom T5000 code
 - Alternate strings
 - Errors
 - Alternate popup dialogs
- These screens are marked in red on the web page as, "AHK Not Supported"
- The current list is:
 - AO-000060
 - Change the string that is displayed
 - DlgLISAccess.cpp ~Line# 77
 - LD-000019
 - Always return sText
 - DlgDesignLabels.cpp ~Line# 3556
 - BRV-00001
 - Force export of HRV.html
 - Apply stash "Force BRV_Lang"

JavaScript

Current Version: HAPS.js :: v1.0a :: 647 lines :: 2025-05-06

Summary:

HAPS has a webpage that gives the user a graphical interface to select the needed screens. This interface relies on JavaScript to create the entire page. The JavaScript will use an array of screen names to populate the categories, screens, menus, and main window. It will also handle adding specific notes or alerts for screens that may require additional steps (i.e., a thumb drive inserted for the “Export Designs” button). Lastly, it will build a single string of the selected screens that is the exact syntax for the screens array that needs to be added to the AHK file (“AHK string”).

Code Architecture:

The code has no main or loop, and uses independent functions to create the page, navigate the page, and update the “AHK string”. The code relies solely on the array of screen names (not screen IDs) to create the page. The code also operates off the assumption that every screen in the array has an associated PNG in the screenshots folder. The start() function is called when the webpage is opened and will create the page.

Code Outline:

- _code array (holds the selected screens)
- _screens array (holds every screen)
- start()
- Navigation Functions
- Code (“AHK string”) Functions
- Alert Function
- Page Creation Functions
- Screen Name/ID Conversion Function
- “AHK Not Supported” Function

Updating Code to Add Screens:

- Ensure the screenshot is in the screenshots folder, in the appropriate subfolder with the correct filename
 - i.e., “screenshots/AO/AO-000001_English.png”
- Create a **unique** screen name
 - Prefix is always the class (i.e., “DlgAdminOptions_”)

- Suffix should be the (easy to understand) identifier for the screen (i.e., “ScreenCleaning”)
 - Another Suffix can be added for more in depth menus (i.e., “DlgAdminOptions_Clean_Success”)
- Add this screen name to the `_screens` array **in the appropriate category**
 - A new category can be added (in the appropriate place) by adding “#Category_Name” to the `_screens` array
- Also add the screen name to the `convertID(screen)` function near the bottom of the code
 - This will tie the screen name to the official Screen ID
- **IF** there are additional steps required for this screen (i.e., a thumb drive inserted for the “Export Designs” button), then add that code to the `alertScreen(screen)` function in the middle of the code
- **IF** the screen is not supported by the AHK, then add that code to the `canHasAHK(screen)` function at the bottom of the code

HTML

Current Version: HAPS.html :: v1.0a :: 42 lines :: 2025-04-16

Summary:

Of course (as usual) the HAPS webpage is only 23 lines of actual HTML :) This file also includes 19 lines of page level CSS. The HTML is only creating the title and divs and their respective styles.