

Staging

An easy to use Android solution for staging Handheld devices and preparing them for production use. You can use a barcode scan or NFC tags for simple profile creation and easy device configuration.

Project

A project contains one or more commands as well as some settings. You can work on multiple projects at any time.

At the top left you will see the title and command count of the current project.

Use the *Open project* button to switch between projects or use *Create or import project* to create a new project or to open a project from a .hsp file.

Demo

2 commands

Open project

+ Create or import project



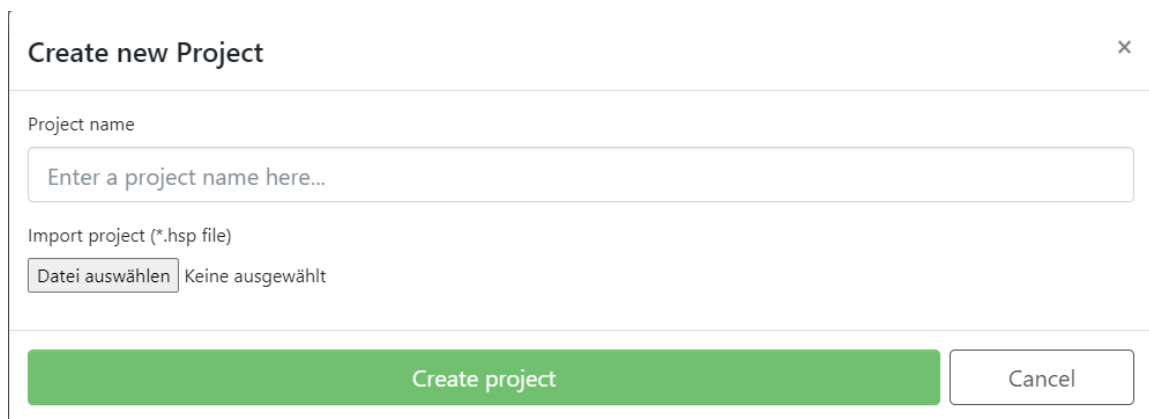
Attention

All projects are only be stored locally in your browser! If you uninstall your browser or clear the browser data, you might lose your projects.

Make sure to download your project files before switching your browser or PC!

Create new project

Use the *Create or import project* button to create a new project or to open a project from a .hsp file.



The screenshot shows a dialog box titled "Create new Project" with a close button (X) in the top right corner. Inside the dialog, there is a section labeled "Project name" with a text input field containing the placeholder "Enter a project name here...". Below this is a section labeled "Import project (*.hsp file)" with a button labeled "Datei auswählen" and the text "Keine ausgewählt". At the bottom of the dialog, there are two buttons: a large green button labeled "Create project" and a smaller white button labeled "Cancel".

Here you can enter a name for your project.

You can also open an existing project by selecting a .hsp file.

If you open a project file, the name will be read from the file and set automatically.

You can still change the name afterwards.

Working with a project

On the top of each project, you have 3 buttons.



The first button opens the command picker where you can choose a new command to add to the project.

The center button action allows you to export the project in different formats.

See [here](#) [#project/export:] for more details on the export options.

The last button lets you rename, clone or delete the project as well as open the [project settings](#) [#project/project-settings:].

Project settings

This panel provides access to the project and barcode settings.

Project settings

Description

☐ Require internet connection

☐ Require WiFi connection

☐ Check for updates to the staging app before running

☐ Require Charging

☐ Continue execution if an error occurs

This can cause an incomplete device setup!

Barcode settings

Bytes per block

300

Barcode type

Datamatrix

Barcode scale

1.5

Cancel

Save

The first section lets you define device requirements for this project. If one of the requirements is not met, the staging will stop with an error message.

- *Require Internet connection* checks for an internet connection of any kind

Export Formats

Export as barcode

This will open a new window with one or more Datamatrix 2D barcodes which contain all the commands. You can read those barcodes using the camera or 2D scanner on a device.

Export as remote script

This will download a JSON file containing the commands. You can place this on your server and trigger it from a different set of commands. This allows you to reduce the size of the initial barcode and to modify the commands on your server depending on the request parameters like the device model or serial number.

Export as binary

This will download a bin file containing the commands in a compressed format. You can use this file to write the commands to an NFC tag with the [TagWriter App](#) [#advanced/tag-writer:].

Export as project file

This will download an hsp file containing the commands and some metadata. You can use this file to import the same project on a different PC or browser.

Commands

A command is a single part of a project that is executed in order when running a project on the device

Adding a new command

To add a new command, press the green *Add command* button at the top of your project.

This will open a window listing all available [command types](#) [#command-types/index:].

Use the search bar at the top to find the command you want to add.

Add new command

Search for command...

Run remote script
Run a remote script to execute commands

Download
Download a file

Install app
Install an app from a URL or local file

Set as home
Set an activity as the default home/launcher

Intent
Start an Android intent

Settings
Change system settings

Copy file
Copy a local file

App permission
Change app permissions

Set wallpaper
Set wallpaper to a local file

Internal script
Run a built-in set of commands

WLAN
Connect to a WLAN network

Delete file
Delete a local file

Homescreen shortcut
Add any app to the home/launcher

Create APN
Create a new Access Point Name






Install handheld app
Install the latest version of a handheld app

Select any command by pressing the *Add [...] command* button.






Editing a command

On the left half of the main panel, you see a list of all your current commands.

The first column shows the position in the list, the 2nd column shows the name of the command with the command type after that.

2	Install KioskHome	app-internal					
---	-------------------	--------------	---	---	---	---	---

At the end of the row, you will find the following actions:

-  opens the command in edit mode. Your changes will not be saved until you hit SAVE at the bottom of the command editor
-  creates a copy of the command, including all the settings, below the current command
-  removes the command
-  moves the command up by one
-  moves the command down by one

The actual editor panel depends on the command type. You can read more about the various command types below.

Command types

On the left you will find a list of all command types with some description and examples.

Some commands are more complex than others so make sure to read the description carefully and feel free to contact support if anything is unclear.

Run remote script

This allows you to run any set of commands from a server. Export a project as a "remote script" and point the command to that file. Currently, only http/https links are supported. You can also use placeholders to provide more details about the device in the request. This can be used to provide slightly different commands to different products or single devices.

For a list of all supported URL schemes, see [here](#) [#advanced/supported-url-formats:]. For possible placeholders you can use in the URL, see [here](#) [#advanced/placeholders:].

Example: Read and execute commands in a script from a server. The device serial number is used as the file name to provide a different configuration for each unit.

Command settings

URL

Save

Cancel

Settings

With this, you can change any setting in Android within the Global, System or Secure namespace.

There is a list of common settings to choose from but you can enter key by choosing the “other setting” option.

For most of the predefined settings, you can only choose ON or OFF but some have a text input. Please make sure to use a possible value in those cases.

For a manual key, you can find a list of possible keys for each namespace [here](https://developer.android.com/reference/android/provider/Settings) [https://developer.android.com/reference/android/provider/Settings].

You can also get the current values from a device using USB-Debugging and the command “*adb shell settings list global/system/secure*”.

Please note that some values might affect the system immediately while others will require a system reboot.

Example #1: Disable the screen rotation using a preset

Command settings

Type

System

Preset

Automatic screen rotation

Value

Off

☐ Is list

Enable this if you want to append your value to an existing list

Save

Cancel

Example #2: Disable the notification that an open Wi-Fi network is nearby

WLAN

Use this to connect to any Wi-Fi network. You can either just set up the network or immediately connect to it.

For the latter case, the staging process will hold until a connection is established.

Example: Connect to a WPA2 secured network

Command settings


Network Name (SSID)

Handheld

Security Type

WPA2-PSK ▼

Password



Password needs to be between 8 to 63 characters

☐ Use proxy?

☒ Connect to this network

The rest of the commands will stop until a connection is established

Save Cancel

Download

With this command, you can download any file to the device. Make sure to provide a valid local path that you can write to. To ensure this, you should use the **%sdcard%** or the **%temp%** placeholder (e.g. "%sdcard%/your-filename.xml").

For a list of all supported URL schemes, see [here](#) [#advanced/supported-url-formats:]. For possible placeholders you can use in the URL or path, see [here](#) [#advanced/placeholders:].

Please note that other apps might be able to access the file at %sdcard% but no other app can access files at %temp%.

A file in %temp% will be available for the whole time while your commands are executed but might be removed automatically after your project is completed.

Since v1.5.0 you can also omit the filename for the destination and only provide a folder. In this case, the destination must end with slash! E.g.: URL="https://example.com/folder/file.jpg" and Destination "%sdcard%/myfiles/" will download the file to "%sdcard%/myfiles/file.jpg".

Example: Download an image to the internal storage with the name "wallpaper.jpg". If the file exists, it will be overwritten by the new download (we'll set this file as the [device wallpaper](#) [#command-types/set-wallpaper:] later)

Command settings

URL

URL to file starting with http(s)://. Can contain placeholders.

Destination

Local file destination. Can contain placeholders.

☒ Overwrite existing file

Copy file

Use this to copy a local file from one path to another. Make sure that both the source and destination are valid paths and that the source does exist.

Example: We copy the file from the previous step to a new location. If the target file already exists, it will be overwritten by the new file

Command settings

Source

Local file source. Can contain placeholders.

Destination

Local file destination. Can contain placeholders.

☒ Overwrite existing file

Save

Cancel

Delete file

Use this to delete any local file on the device. The command will not fail if the file doesn't exist or can't be deleted!

Example: We delete the original version of the copied file

Command settings

Path

Local file path. Can contain placeholders.

Save

Cancel

Install app

With this command, you can install any app from a local or remote apk file.

For remote files, only http and https is supported. A remote file will be downloaded first so please ensure a necessary connection before this command.

For a list of all supported URL schemes, see [here](#) [#advanced/supported-url-formats:]

Example: We install an app from a server. (This will download the file to a temporary location before installing it)

Command settings

Path

Local path or URL starting with http(s)://. Can contain placeholders.

Save

Cancel

App permissions

This command enables you to grant or revoke any permission on an app. This is the same as changing Settings > App > (your app) > Permissions.

An app will usually ask for it's required permissions on demand or when it's launched for the first time. With this, you can grant those permissions beforehand.



Hint

To grant all permissions that a apps has defined, use "all" as the *App permission* value

Example: We allow the app with the package name "com.handheldgroup.someapp" to read the phone contacts

Command settings

App package

App permission

Type

Save

Cancel

Homescreen shortcut

Add any app to the home screen. This must be supported by the launcher app and will cause an error otherwise!

To find the package name of an app, you can use a tool like [List My Apps](https://play.google.com/store/apps/details?id=de.onyxbits.listmyapps) [https://play.google.com/store/apps/details?id=de.onyxbits.listmyapps].

Example: Add Chrome to the home screen

Command settings

This command might work differently depending on the device and Android version. It will always add an icon but the actual location might not correspond to the input!

Package name / Class

The full classpath of the activity or the package name. If you provide a package name, the default activity will be used

Screen

Position of the screen, starting with 0, where you want to add the icon. Use -1 to add the icon to a new screen

Icon column

Column to use on the screen, starting with 0, where you want to add the icon. Use -1 find the next free spot

Icon row

Row to use on the screen, starting with 0, where you want to add the icon. Use -1 find the next free spot

Set as home

If you install any kind of home/launcher replacement, you can set that app as the default launcher with this command.

Make sure to provide the full classpath to the launcher activity.

Example: We set the launcher app of the KioskHome as the default home/launcher app

Command settings

Class

com.handheldgroup.kioskhome/com.handheldgroup.kioskhome.activity.HomeActivity

Save

Cancel

Set wallpaper

With this command, you can set any local image file as the wallpaper. Make sure to provide an existing local path to an supported image.

Example: We set the downloaded image as the wallpaper

Command settings

Path

Local path to image file. Can contain placeholders.

Save

Cancel

Create APN

This function allows you to create an APN. Make sure to check the “Set this APN as preferred” option at the bottom to actually enable this APN.

Most standard provider APNs are already included in the system and should appear as soon as a SIM card is inserted. You’ll only have to use this commands if you need a special APN for your use case.

Intent

This command allows you to start an Activity (by calling `startActivity`) using an Intent (classpath) or Action. You can provide a set of extras to pass to the activity as well.

You can use this to trigger an action or initialization in your app.

Make sure that the called Activity does actually exit after it's done because the staging will hold until the activity stops.

When starting an Activity, there are three possible ways to provide the "Intent value":

- The full class path to the activity you want to start – e.g. `"com.android.chrome/com.google.android.apps.chrome.Main"` for Chrome
- The package name suffixed by a slash to start the default/launcher activity inside that app – e.g. `"com.android.chrome/"` for Chrome
- A intent action to start any app associated with that action – e.g. `"android.intent.action.VIEW"` to view a element. If you sent the data to a URI, this can also open Chrome

To set the data element in the intent, add a string extra and use the special value `"__DATA"` as the key for this extra

You can also use intent commands to apply settings to other handheld and MaxGo app. See [here](https://developer.handheldgroup.com/docs/handheld_common/config-intents/) [https://developer.handheldgroup.com/docs/handheld_common/config-intents/] for details and supported apps.

Internal Script

This allows you to run a predefined set of internal commands. This is built from other commands and is mostly used to keep the total size of the export smaller.

Example: We run the necessary steps to set up the KioskHome app.

Make sure to only run this AFTER installing the app

Command settings

Script

Set up KioskHome as the default launcher ▼

Save

Cancel

Install handheld app

This will automatically download and install the latest version of the selected app for the device. Please note that some of our apps might not be available for all devices. The command will fail with an error if no match is found.

Example: We install the latest version of the KioskHome app

Command settings

App

KioskHome ▼

Save

Cancel

Appops settings

With this command, you can run “appops” commands granting your apps special permissions that usually must be allowed by the user manually. This includes settings like “Change system settings” or “Draw over other apps”.

Example: We allow the app with the package name “com.handheldgroup.someapp” to draw over other apps

Command settings

App package

com.handheldgroup.someapp

AppOps operation

SYSTEM_ALERT_WINDOW

Type

allow

Save

Cancel

Package Manager

This command allows you to call the package manager and allows you to uninstall, hide or reset installed apps.

Please note that uninstalling system apps will fail. For a similar result, you can use “hide”.

Example: Hide the pre-installed Chrome browser (package name “com.android.chrome”) so the user can’t open it anymore

Command settings

Operation

Hide ▼

App package

com.android.chrome

Save

Cancel

Language

Use this command to change the system language and region of your device.

Example: Change the language to German and the region to Germany

Command settings

Language

German (Germany) ▼

Please note that not every Android version will support all the options above. Android will try to fall back to the best option if the selected is not available

Save

Cancel

Self-update

This command triggers an update check for the staging app.

If you want to run this check before any other commands, you can also use the option in the [project settings](#) [`#project/project-settings:`].

Running this command at a later time is useful if you need to set up an internet connection first.

NTP Server

This command allows you to change the time server that Android will use to query the current time and date. This is only useful if your network doesn't allow access to the default Android time server.

If the device has a cellular connection, it will also try to pull the date information from that network and will usually prefer that information to the NTP server.

Lockscreen

Change the lockscreen type.

Example: Set the lockscreen to use a PIN "1234"

Command settings

Security

Pin

PIN

1234

Numeric PIN

Save

Cancel

Sidekey

Change the behavior of a side-/function key.

The keys you can change depend on the device. See [here](https://developer.handheldgroup.com/docs/staging/side-keys/) [https://developer.handheldgroup.com/docs/staging/side-keys/] for possible keys.

Example: Set the F1 key to trigger the scanner

Command settings

Side key name

Name for the side key. See [here](#) for possible values.

Keycode

The keycode or action to trigger on this key

Save

Cancel

User restriction

Enable or disable a user restriction which allows to lock down certain functionality of the device.

Example: Prevents the user from making outgoing calls

Command settings

key

Disallow outgoing calls

Specifies that the user is not allowed to make outgoing phone calls. Emergency calls are still permitted. The default value is false. This restriction has no effect on managed profiles.

Value

True

Save

Cancel

System property

Change a system property via the "setprop" command.

Install user certificate

Add a user certificate to the Android certificate store. You can use a remote or local *Path*.

For a list of all supported URL schemes, see [here](#) [#advanced/supported-url-formats:]

Send metadata

This command allows to collect information about the devices you are running the project on.

This can be usefull if you need all the Wi-Fi MAC adresses to allow in your network or to collect all the IMEI numbers.

You can either use our provided server by using the "MaxGo Staging Meta" preset.

If you use our server, make sure to save the "Meta UUID" somewhere since you will need this value later to retrieve your data [here](https://staging.maxgo.io/editor/meta) [https://staging.maxgo.io/editor/meta].

If you use the "Custom" preset, you send the data to your own server.

For the values, you can use a comma-separated list of [placeholders](#) [#advanced/placeholders:]. The serial number of the device will always be collected.

Placeholders

You can use various placeholders to provide details about the device on remote calls and to define device independent paths for download.

Key	Name	Note
%temp%	Temp folder	Temp / Cache folder on the device. Ideal location for files that you don't need anymore after staging is done
%sdcard%	Internal storage	Internal storage folder on the device. Ideal location for files that should be accessible to other apps or will be needed after staging
%ext-sdcard%	Removable SD card	Path to a removable SD card. If none is present, the internal %sdcard% path is returned
%hw-model%	Device model name	Name of the device hardware/product
%hw-os%	Device OS version	Current Android OS version / SDK level
%hw-version%	Device OS version	Current OS build version as shown in Settings > About
%hw-sn%	Device Unit S/N	Unit serial number. Some devices can have two serial numbers. This will always return the S/N used on the label and box
%sn%	Device S/N	Device serial number
%version%	App Version	Current Version of the Staging app
%version-code%	App Version code	Current numerical version code of the Staging app
%_t%	Timestamp	Current time as a timestamp. Useful to

Supported URL formats

Any command taking a URL (download, install, remote-script, etc) supports the following schemes:

http

`https://www.example.com/path/to/file.apk` will perform a HTTP or HTTPS GET request to the URL. You can change the port by passing “:4200” after the host.

Authentication is supported by providing the username and password as part of the URL.

e.g.: `https://username:password@www.example.com/path/to/file.apk`

ftp

`ftp://user:password@example.com/path/to/file.apk` will login to the FTP server and try to download the file at the given path. If user and password is omitted, an “anonymous” login is performed. You can change the port by passing “:4200” after the host.

Format is `ftp(s)://[user[:password]]@server[:port]/path/to/file.ext`

smb (Samba / Windows Network Share)

`smb://user:password@server/path/to/file.ext` will try to access the network share on “server” with the provided login.

Format is `smb://[domain]user[:password]@server/path/to/file.ext`

File hosts

MaxGo Staging v1.8.0 supports downloads from some file hosts that don't serve a file directly on the initial URL. Make sure to always share the file with "everyone" as Staging does not support authentication

Metadata

The page allows you to retrieve the data collected via a [send metadata](#) [#command-types/send-metadata:] command by providing the "Meta UUID" that was used by the command.

Staging Tag Writer

The Staging Tag Writer app allows to write a Staging project to a NFC chip.

You can use the app on any Android device. The app can be downloaded on the [PlayStore](https://play.google.com/store/apps/details?id=com.handheldgroup.staging.tagwriter) [https://play.google.com/store/apps/details?id=com.handheldgroup.staging.tagwriter] or directly as an APK from [here](https://apps.maxgo.io/staging-tagwriter) [https://apps.maxgo.io/staging-tagwriter].

The app works with any NDEF compatible NFC chip that has user-writable storage.

You can directly scan the barcode from the Staging editor or use a binary export.

Sidekeys

The SidekeyCommand allows you to change the functionality of the side keys on supported devices. You can change the keycode or launch an app.

The name you have to use for each button differs from product to product.

However, there will be no error if you use an invalid name and there shouldn't be any conflicting names. This means that you can still use the same staging project across multiple products.

Please find the actual names for the different buttons below:

Nautiz X2

Older versions of the Nautiz X2 OS require a reboot for the settings to take effect!

Name	Description	Note
left-up	Left upper side key	
left-down	Left lower side key	
right-up	Right upper side key	
right-down	Right lower side key	
front-scan	Front scan trigger (3rd from the left)	
gun-scan	Trigger on the pistol grip	
front-home	Front Home button (1st from the left)	You can not start an app with this button!
front-menu	Front Menu/Recent apps button	You can not start an app

Scanner settings

Please note that this feature requires at least v1.6.4 of the Staging app!

This document only applies to the Nautiz X2 and Nautiz X41. It's currently not possible to change the scanner settings on other devices!

Each settings is applied using a [Intent](#) [#command-types/intent:] command. If you want to change multiple settings of the same type (e.g. Symbolologies) you will have to create a Intent command for each change.

The Intent Type is always "Action".

Enable and Disable Symbolologies

Intent value: android.intent.action.ENABLESYMBOLOLOGIES

Extras:

- Type: Integer, Key: "symbolologies", Value: see list below
- Type: Boolean, Key: "enable", Value: "true" to enable the symbology, "false" to disable it

List of "symbolologies" values:

- 0: Aztec
- 1: Codabar
- 2: Code 11
- 3: Code 128
- 4: Code 39
- 6: Code 93
- 8: Data Matrix

How to open a website in Browser/Chrome

Please note that this feature requires at least v1.4.1 of the Staging app!

1. Create a new Intent command
2. Set the "Intent Type" to "Activity"
3. Set the "Intent value" to "android.intent.action.VIEW"
4. Click "Add extra"
5. Set the "Type" to "String"
6. Set the "Key" to "__DATA"
7. Set the "Value" to the URL you want to open - e.g. "https://example.com/"

Config intents

Usage

Config intents are a way to change the setting of some handheld apps from your own app. The system works by using broadcasts with the actual data provided as extras.

Broadcast action

The broadcast action consists of the apps package name with “.WRITE_CONFIG” appended.

For the KioskBrowser, this would be
“com.handheldgroup.kioskbrowser.WRITE_CONFIG”.

Broadcast extras

You must provide all the different options you want to apply as individual extras.

The available values might differ greatly between apps. Please select one of the [supported apps](#) [#advanced/config-intents/supported-apps/index:] to see possible keys for the app.

Supported apps

The following apps are supported by config intents:

- [MaxGo KiosHome](#) [#advanced/config-intents/supported-apps/kioskhome:]
- [MaxGo KiosBrowser](#) [#advanced/config-intents/supported-apps/kioskbrowser:]
- [MaxGo RFID](#) [#advanced/config-intents/supported-apps/rfid:]
- [MaxGo RTK](#) [#advanced/config-intents/supported-apps/rtk:]

KioskHome

Intent action

`com.handheldgroup.kioskhome.WRITE_CONFIG`

Possible values

Type	Key	Value	Note
String	password	1234	Set password to “1234” (or any other numeric value)
Boolean	show_notifications	true/false	Show notifications inside the app
Boolean	block_statusbar	true/false	Prevent access to the status bar
Boolean	block_recent_apps	true/false	Prevent access to the recent apps list
Boolean	ignore_default_apps	true/false	Do not enable the default apps (phone and contacts)
Boolean	set_allowed_kiosk.internal.wifi	true/false	Show the

KioskBrowser

Intent action

`com.handheldgroup.kioskbrowser.WRITE_CONFIG`

Possible values

The application let's you export the values to a xml file. Please look at that file for the possible keys.

RFID

Intent action

`com.handheldgroup.rfid.WRITE_CONFIG`

Possible values

Type	Key	Value	Note
String	module	NX2EXP-LF	For possible values see “Options for module extra” here
String	password	1234	Password for the app settings
Boolean	auto_start	true/false	Start RFID service when device boots
Boolean	rfid_trigger	true/false	Trigger scan with F7 key (Please see your device manual on how to assign a button to act as F7)
String	output_method	See note	“keyboard”, “paste”, “broadcast”, or “clipboard”
String	terminator	See note	“none”, “enter”, “tab”, or “space”
Boolean	rfid_ack_tone	true/false	Play sound after successful scan
Boolean	show_scan_toast	true/false	Show toast with scan result after successful scan
String	data_prefix		Prefix text
String	data_suffix		Suffix text

RTK

Since RTK app version v1.3.3

Intent action

`com.handheldgroup.rtk.WRITE_CONFIG`

Possible values

Type	Key	Value	Note
String	host	See note	Ntrip server IP or DOMAIN
String	port	number	Ntrip server port
String	mountPoint	See note	Ntrip server mountpoint
String	username	See note	Ntrip client username
String	password	See note	Ntrip client password
String	tcp_port	4444	Device TCP server port
String	antenna_height	120	Antenna height (vertical offset in cm)
Boolean	mock_location	true/false	Enables mock location (not available/necessary on RT10)
Boolean	active_led_x6	true/false	Enables LED indicator for gnss fix status
Boolean	auto_start	true/false	Auto starts app after reboot
Boolean	auto_connect	true/false	Auto starts ntrip connection (auto start must be true!)