

# ekey bionyx Control4 Notification Server (ekeybionyxC4N - version 4)

---

The ekey bionyx Control4 Notification Server (ekeybionyxC4N) is a server driver designed to run on the Control4 controller. Its primary function is to receive and process HTTPS push notifications from ekey devices (fingerprint scanners or control units) via the bionyx app. These notifications enable event triggering based on specific user actions.





The driver is event-focused: it receives data, triggers an event, and allows the system integrator to define the event's functionality in the Control4 Composer Programming section. The driver's scope is limited to generating and triggering events. What an event executes or performs after being triggered depends on the system integrator's own programming such as checking time of the day to execute a scene adjusting light/temperature/shades in a room, arming/disarming security system, unlocking a door, etc. The driver is one-way (receive-only) and does not send commands or requests back to the ekey bionyx app.

## Key Features:





1. Fingerprint Activity:
  - When a valid fingerprint notification is received, the system triggers an associated event based on the scanner ID, user ID, and finger index.
  - During setup, you can assign specific events to individual fingers or have multiple fingers share the same event.
  - A general event is available for invalid and unrecognized scans.
2. Input Activity:
  - When a valid digital input notification is received, the system triggers an associated event based on the controller ID and input number.
  - If an input number is unrecognized, a general "unknown input" event per device is triggered.

## Benefits of Integration:

ekey bionyx integration enhances Control4's powerful event and scene customization by bringing an Adaptive Experience to each user through the secure triggering of events. This transforms Control4 from static automation into a dynamic, intelligent system that personalizes the smart home experience based on:






-  **Who** is triggering the event
-  **Which finger** they scanned
-  **Time of day** they scanned their finger
-  **Location** within the home where they scanned their finger

Imagine how ekey bionyx integration enhances everyday life for a family using Control4:

-  **Dad's Control 4 Adaptive Experience**  
When Dad arrives home, Control4 announces, "Dad is home," while adjusting the AV and AC in his office to his personal settings.
-  **Mom's Control 4 Adaptive Experience**  
As Mom enters, Control4 welcomes her with a custom announcement and activates her preferred lighting and music.
-  **Kids' Control 4 Adaptive Experience**  
When the kids unlock the door, Control4 recognizes them and automatically sends a notification to Mom and Dad—letting them know their kids are home or alerting them if they're late.
-  **Enhanced Security with Control 4 Adaptive Intelligence**  
If anyone in the family feels threatened, they can scan an unusual finger (e.g., their middle finger), triggering a silent alarm—adding an extra layer of security beyond standard automation.

ekey bionyx integration transforms Control4 into a system that doesn't just automate—it understands. Whether it's recognizing family members, service personnel, or unexpected actions, it delivers an intelligent, secure, and personalized experience like never before.

This document includes the following topics.

-  [ekeybionyxC4N Scope and Requirements](#)
-  [ekey bionyx app Configuration and Setup](#)
-  [Driver Configuration and Setup in Composer Pro](#)
-  [Technical Support](#)
-  [Change Log](#)

## ekeybionyxC4N Scope and Requirements

---

### 1) **Control4 Controller and Composer PRO Access:**

- a) A Control4 controller is essential for the operation of the ekeybionyxC4N driver.
- b) Composer PRO access is necessary for configuring the ekeybionyxC4 driver.

### 2) **ekey bionyx Account and bionyx App:**

- a) An ekey bionyx account with System Administrator rights and firmware version 5.2 or higher is required.
- b) Activation of the Notification API and Show push notification in the bionyx app by the system administrator.

### 3) **License Requirements:**

Once added to the project, the ekeybionyxC4N driver runs in demo mode for 5 days, supporting up to 5 active users and 2 input-output extensions per controller. After the demo period, a valid User License is required to continue using the driver.

- **Users License:** A Users License can be purchased as either a new license or an upgrade. Licenses are available for 20 to 100 users, in increments of 10.
- **IO Extension License:** The basic system includes two input-output extensions per controller. If additional controller extensions are needed, a separate IO Extension License is required. New and upgrade licenses are available for 2 to 18 extensions, in increments of 2.

To purchase a User License or an IO Extension License, please contact [License@ekeyUSA.com](mailto:License@ekeyUSA.com) with:

- Your contact details
- Control4 MAC address
- Number of active users and input-output extensions needed.

Properties

PropertiesSummary

Properties

Properties

Actions

Documentation

Lua

Debug Mode

Detailed Debug

Driver Version

4

Control 4 URL

192.168.254.60

Controller MAC Address

000FFF527A57

ekey bionyx Listening Port

58000

Driver Activation Notice:

This driver operates in demo mode for 5 days from the date of installation.

After the demo period, a valid license is required to ensure continued functionality.

To purchase a license and activate this driver, please contact us at [license@ekeyusa.com](mailto:license@ekeyusa.com).

Include your contact information, Control4 MAC address, and the desired number of users and Input-Output (IO) Extensions if required.

bionyx System Name

Family System

Users License

Demo Remaining Time: 4 days, 23 hrs, 44 min, 44 sec

Controller I/O Extensions Li

Default: 2 Input-Outputs, 0 Extensions

Properties

PropertiesSummary

Properties

Properties

Actions

Documentation

Lua

Debug Mode

Detailed Debug

Driver Version

4

Control 4 URL

192.168.254.60

Controller MAC Address

000FFF527A57

ekey bionyx Listening Port

58000

Driver Activation Notice:

This driver operates in demo mode for 5 days from the date of installation.

After the demo period, a valid license is required to ensure continued functionality.

To purchase a license and activate this driver, please contact us at [license@ekeyusa.com](mailto:license@ekeyusa.com).

Include your contact information, Control4 MAC address, and the desired number of users and Input-Output (IO) Extensions if required.

bionyx System Name

Family System

Users License

U050- -5364 ==> License is Valid ==> 50 Users

Controller I/O Extensions Li

IO08- -9E4D ==> License is Valid ==> 8 Input-output

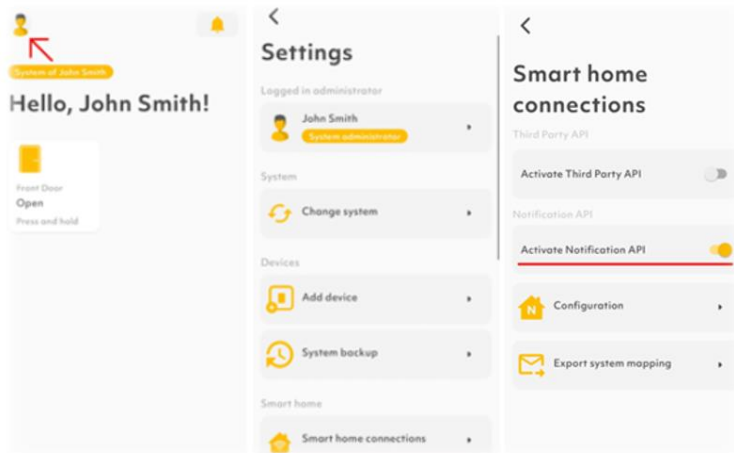
## ekey Bioynx app - Configuration and Setup

### 1. ekey bionyx Account and bionyx App:

- An ekey bionyx account with System Administrator rights and firmware version 5.2 or higher is required.

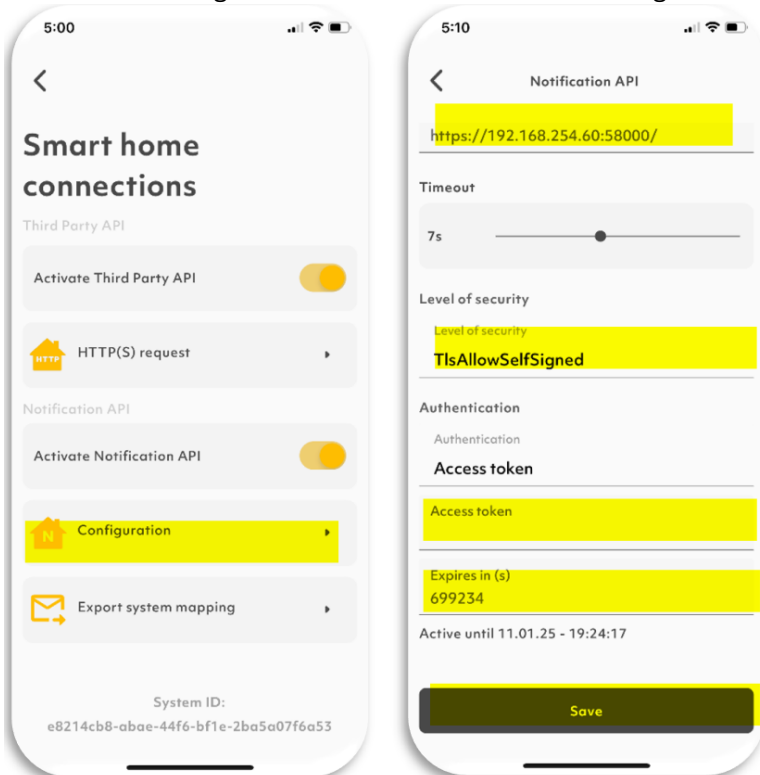
2. Activate the Notification API in the bionyx app:

- Go to Home > Settings > Smart Home Connections, then activate the Notification API.



1. Notification Webhook API configuration

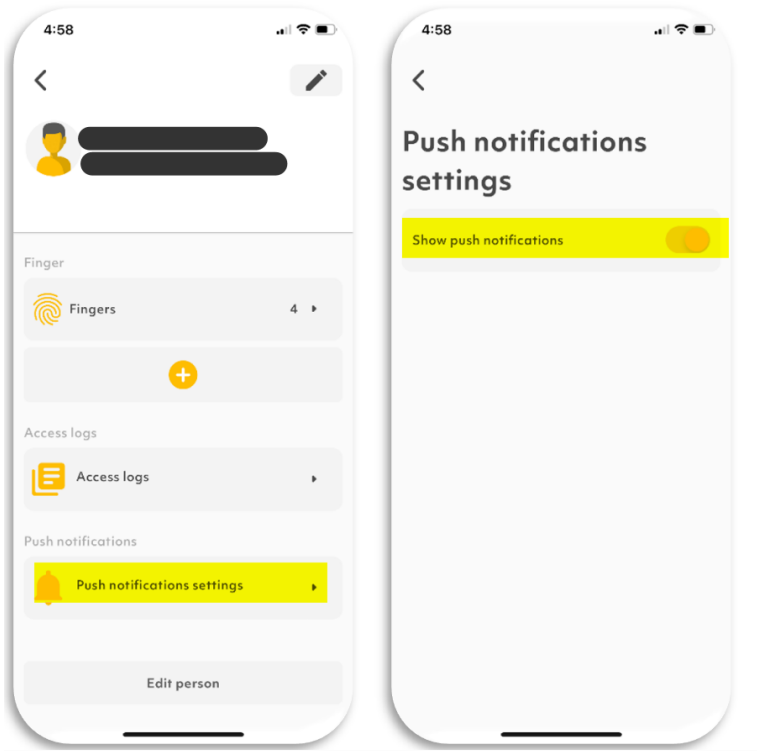
- a. Go to Home > Settings > Smart Home Connections > Configuration, then update shown fields as follows:
  - i. URL: Type Control4 URL following this format: [https://Control4\\_IP\\_Address:Port\\_Number/](https://Control4_IP_Address:Port_Number/) for example “https://192.168.254.200:58000/”
  - ii. Timeout: leave it on 7s (default).
  - iii. Level of Security: click and select "TlsAllowSelfSigned" from the list.
  - iv. Authentication: click and select “Access Token” from the list. Then, type the same code entered in the Control4 driver properties “Access Token” field.
  - v. Expires in (s): type 700,000 seconds.
- b. Saving the Notification Webhook API configuration: Press on “Save” to save the HTTPS request.



2. Activating Push Notification







- a. This will active the Push Notification for all users in the system.

- b. Go to Home > Settings > Logged in administrator>Push notifications setting> then activate Show Push Notification



# Driver Configuration and Setup in Composer Pro

The ekeybionyxC4N driver initial configuration and setup in the Composer Pro composed 5 steps.

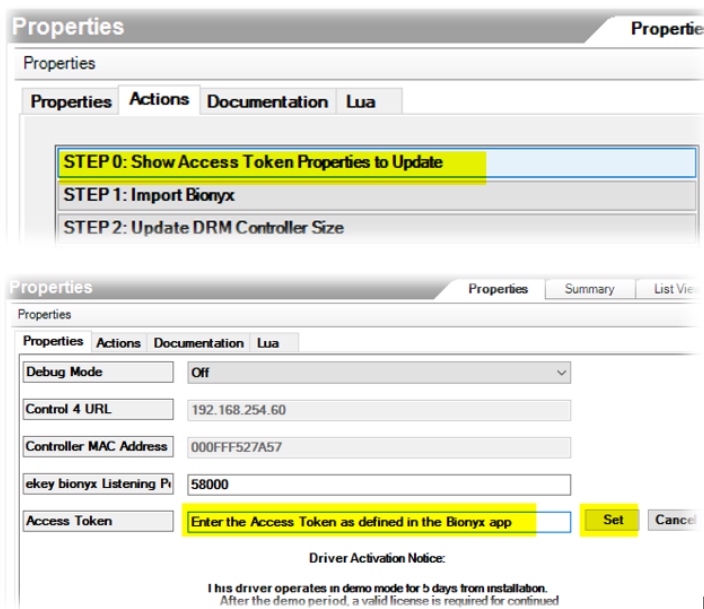
-  STEP 0 (REQUIRED): Show Access Token Properties to Update
-  STEP 1 (REQUIRED): Import bionyx Configuration
-  STEP 2 (OPTIONAL): Update DRM Controller I/O Extensions
-  STEP 3 (OPTIONAL): Update Finger Assignments
-  STEP 4 (REQUIRED): Generate eKey Bionyx Events
-  STEP 5: Manual Programming for Events (Reminder: No Action Required here)

## STEP 0 (REQUIRED): Show Access Token Properties to Update

The driver uses the Access Token code to authenticate HTTPS messages origin. The same access token entered in the ekey bionyx App Settings (Smart Home Connections > Configuration) should be entered here. This is a required step upon initial setup and when the Access Token in the bionyx App and the driver needs to be changed.

Follow these STEPs to setup the Access Token code:

- 1) Click on Actions on the tope menu of the Composer Pro and select STEP 0: Show Access Token Properties to Update. This will show Access Token on the Property screen and enable it for editing.
- 2) Click on Properties on the top menu and type in the Access Token code used in the ekey bionyx App Settings (Home >Settings > Smart Home Connections > Configuration)
- 3) Click on Set to update the Property value.
- 4) Once saved, the driver will hide the Access Token value Property.



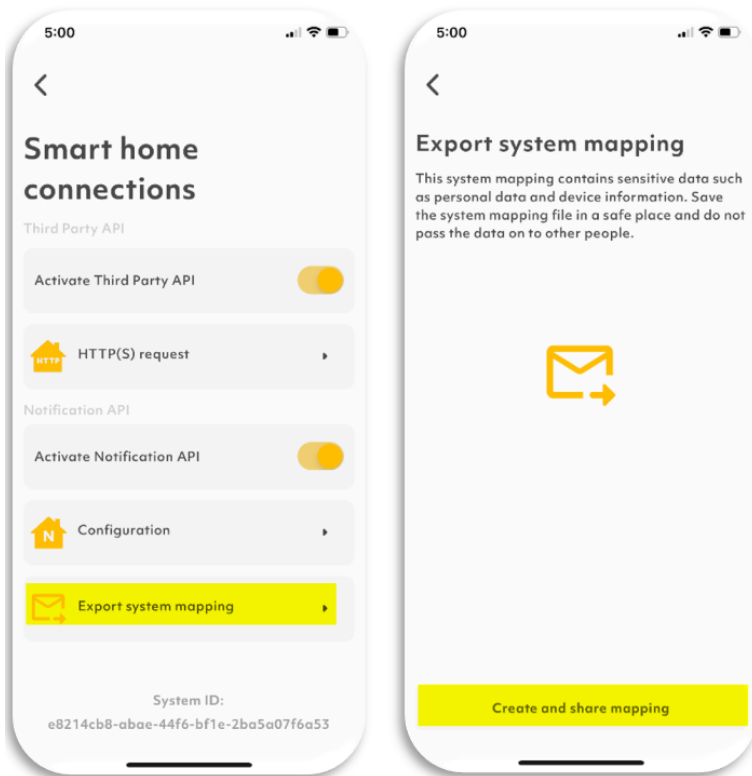
## STEP 1 (REQUIRED): Import bionyx Configuration

The bionyx app allows you to export current users and devices details into the driver using a JSON file named “system\_mapping\_export.json”. This step is required and must be successfully completed before proceeding. Once a valid JSON file is decoded, the system imports the configuration, including the system name, users, and device details. The driver will display the status of this step and any error explanations in the Lua Output Screen.

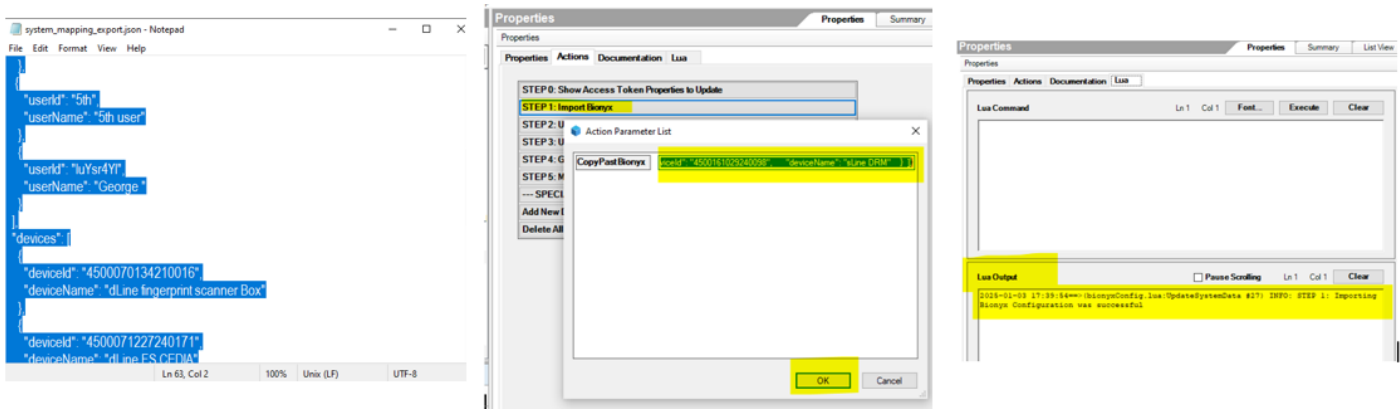
Follow these STEPs to generate and import the file:

- 1) Generate the Configuration File:
  - a) Open the bionyx app on your smart device.

- b) Navigate to Settings > Smart Home Connections > Export System Mapping.
- c) Select Create and Share Mapping and follow the prompts to email the generated “system\_mapping\_export.json” file to yourself or save it directly to your computer’s hard drive.



- 2) Open the Configuration File in Notepad:
  - a) To ensure compatibility, use **Windows Notepad** for opening the JSON file. Avoid using other text editors to prevent formatting issues. Follow these STEPs to open Notepad:
    - i) Locate Notepad: Click the Start Menu (Windows icon in the bottom-left corner). Type Notepad in the search bar and select it from the results.
    - ii) Alternatively, press **Win + R** to open the Run dialog, type **notepad**, and press Enter.
  - b) Open the JSON File:
    - i) In Notepad, go to File > Open.
    - ii) Navigate to where the “system\_mapping\_export.json” file is saved and open it.
    - iii) Click anywhere in the open file, then press **Ctrl + A** and then Ctrl + C to select all and copy text. The copied text will be pasted in the next STEP.
- 3) Import Data into the Driver:
  - i) In Composer, go to the Actions section and select the command STEP 1: Import bionyx.
  - ii) Paste the copied JSON data into the Copy-Paste bionyx field.
- 4) Check the Import Status and verify the completion status or error details in the Lua Output window.



Note:

*Depending on how the JSON file is imported, copying and pasting the data may not display everything correctly due to how carriage returns and line feeds are handled.*

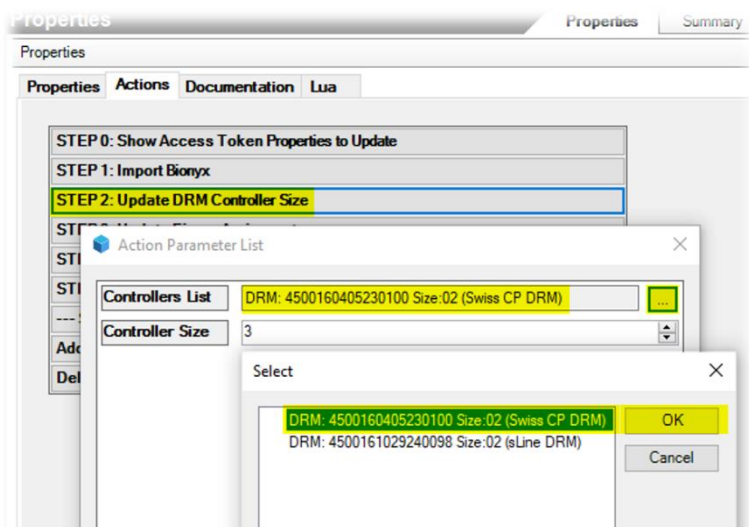
*If some data is missing after pasting, open the attached LForly.txt in Notepad, paste the JSON data into the file, then copy all the content and paste it into the Composer Bionyx field.*

## STEP 2 (OPTIONAL): Update DRM Controller I/O Extensions

ekey offers two types of control units: dLine Control Unit used in the dLine system, it includes 1 relay (common and normally open) and 1 digital input. And DRM Control Unit: Used in the xLine and sLine systems, it includes 2 relays (common and normally open) and 2 digital inputs. The DRM control units support relay extensions, available in two sizes: 2 relays/inputs and 4 relays/inputs.

This step is optional but required if the Bionyx system includes input-output extensions. Therefore, STEP 2 allows you to manually update the size (number of relays/inputs) of DRM controllers as required. The driver generates dynamic list of DRM controllers from the imported bioynx configuration in STEP 1 with capability to update the size of each of these controllers. The driver will display in the LUA Output Screen the status of the controllers' updates and explanation of any errors. Follow these STEPs to update DRM Controller Sizes:

- 1) In the Composer Go to the Actions section and select the command STEP 2: Update DRM Controller Size.
- 2) **Review of the Controller List:** The driver will dynamically generate a list of DRM controllers from the imported data in STEP 1. Each item in the list will include the control type (DRM), the controller's name (as configured in the bionyx app), and its current size (relay/input count). Select the controller you want to update.
- 3) **Adjust Controller Size:** In the Control Size field, specify the desired size (number of relays/inputs). The size can range from 1 to 18, with a default value of 2.
- 4) **Confirm Updates:** Press OK to save changes.
- 5) **Repeat as Needed:** Repeat STEPs 2–4 for each DRM controller in the list that requires adjustment.
- 6) Check the update results and verify the completion status or error details in the Lua Output window.



## STEP 3 (OPTIONAL): Update Finger Assignments

The ekey bionyx push notifications for finger events identify the specific finger responsible for generating the notification. The driver allows you to trigger events based on these actions, offering flexibility to assign unique events to individual fingers or share events among multiple fingers for a user on a device.



Assigning unique events for every finger across all users can lead to thousands of events, increasing system complexity and making event selection cumbersome. To optimize this, the driver supports mapping fingers to either specific events (unique per user, finger, and device) or common events (shared among multiple fingers for a user on a device). Upon completion of this STEP, the driver will display: "STEP 3: Finger Assignments update completed. X finger(s) are linked to specific events." Before proceeding to STEP 4, you may revisit and adjust your selections as needed.

Follow these STEPs to update finger assignments:

- 1) In the Composer Go to the Actions section and select the command STEP 3: Update Finger Assignments.
- 2) **Review of the Finger List:** A generic list of left- and right-hand fingers will be displayed. Each finger will have two options for assignment. Common Event: The finger will trigger the user's shared, common event. and, Specific Event: The finger will trigger a unique, specific event assigned to it.
- 3) **Assign Finger Events:** By default, the driver default assignment for all fingers is the Common Event option. change each finger assignment as desired.
- 4) **Optimize Assignments:** Assign finger-specific events only when necessary to avoid long list of events and unnecessary complexity. Keeping the number of unique events minimal will simplify event selection and programming.
- 5) **Confirm Your Assignments:** Once all finger assignment adjustments are complete, confirm your configuration by pressing OK.
- 6) Check the update results and verify the completion status or error details in the Lua Output window.

## STEP 4 (REQUIRED): Generate eKey Bionyx Events

This is the last required step for generating unique events based on the configuration from previous STEPs, linking users, devices, and fingers to trigger specific actions. The generated events will be accessible in the Composer Programming section so that it can be personalized and programmed as desired. The driver will display in the LUA Output Screen the status of the generation process and provide an explanation of any errors

### Note

*The driver's scope is limited to generating and triggering events. What an event executes or performs after being triggered depends on the **system integrator** and their programming, which is beyond the driver's control.*

*If Step 4: Event Generation is repeated, there is a risk that events may **\*\*inherit previous programming incorrectly\*\*** or that a different event may be assigned to a new event. Due to this technical limitation, event programming should be carefully **\*\*reviewed and verified\*\*** to ensure correct assignments. Special attention should be given to confirming that each event is properly programmed.*

Follow these STEPs to generate ekey bioynx events:

- 1) In the Composer Go to the Actions section and select the command STEP 4: Generate ekey bionyx Events.
- 2) choose one of the following options: YES - Ready to Proceed: Proceed with event generation. NO - Review Configuration: Go back to reviewing the configuration, extensions, or specific event assignments.
- 3) **System Event Creation:** Once you confirm and click OK, the system will generate events based on your setup. The events generated follow this sample **"001) Username -> Finger Name @ Device Name"** with the following naming convention"


 001: A unique numeric ID, listed in ascending order.

 Username: The user's name, as imported in STEP 1 from the bionyx configuration file.

- For unrecognized finger events, an event with "Unknown Finger" will be created for each scanner.
- For unrecognized digital inputs, an event with "Unknown Input" will be created for each controller.

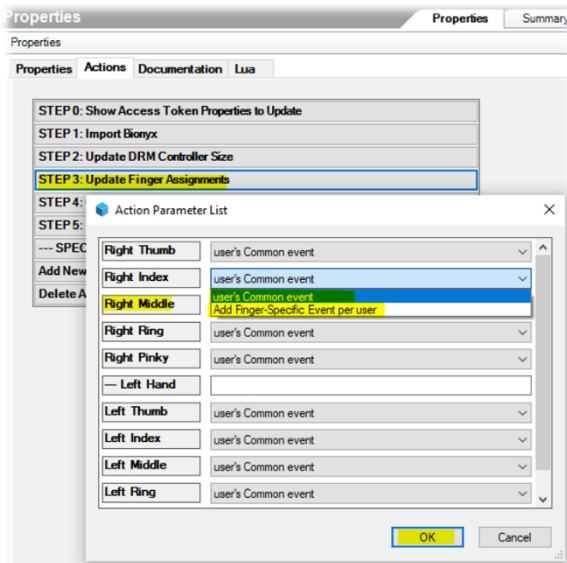
 -> Event Name:

- Finger Events: The specific finger, e.g., "Right Index," or "Any Finger" for common events.
- Input Events: the text "Input " followed by a sequential input number, starting from 01 up to the controller's maximum size, as specified in STEP 2.

 @ Device Name: The name of the scanner (for fingerprint events) or the controller (for digital input events) as imported in STEP 1 from the bionyx configuration file.






**Note:** "Unknown Finger" and "Unknown Input" events are grouped at the end of the event list for clarity and easier reference.

- 4) Check the generation results and verify the completion status or error details in the Lua Output window.



## STEP 5: Manual Programming for Events (Reminder: No Action Required here)

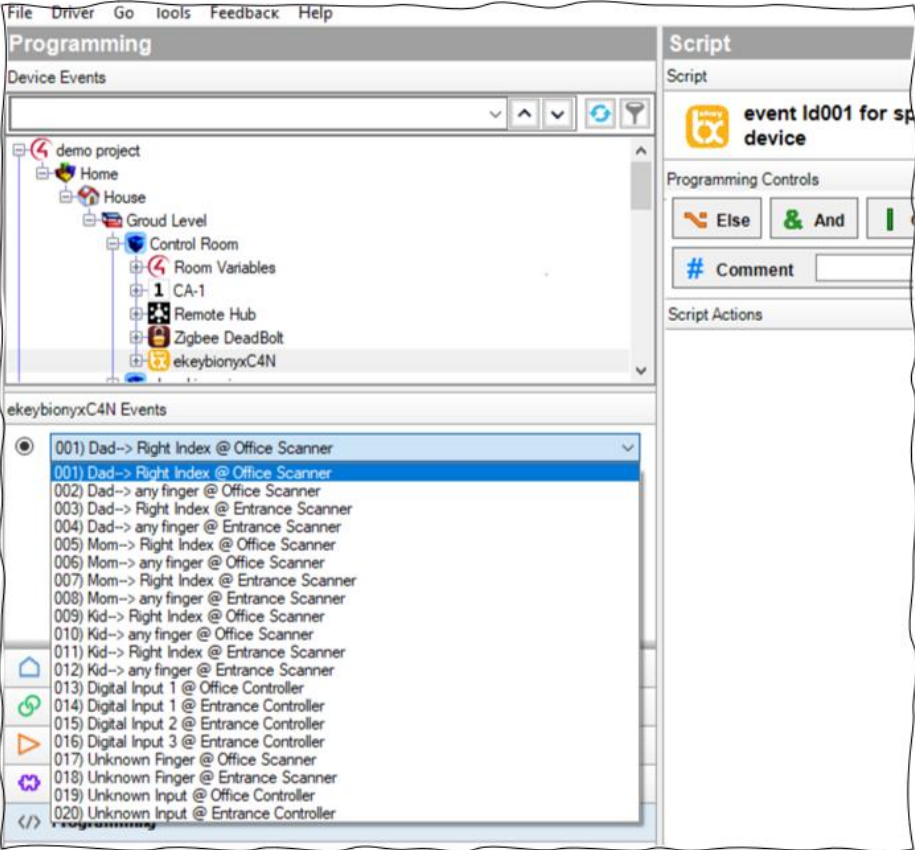
This is the last STEP in the driver's configuration. It involves a manual process performed by the system integrator in the Programming section, where each previously generated event can be programmed and personalized to meet specific requirements. This is where the true integration of ekey devices with Control4 comes to life, enabling customized automation triggered by a user's unique fingerprint. For instance:

-  Opening doors by unlocking any Wi-Fi, Z-Wave, Zigbee, or Bluetooth wireless lock accessible through Control4, ensuring seamless and secure access.
-  "Welcome Home" and "Goodbye Mode" can adjust lighting, security, and temperature based on the family member's fingerprint.
-  Notifications for events like kids returning from school or package deliveries keep homeowners informed.
-  Emergency features, such as panic buttons and lockdown modes, enhance security.
-  Automated entertainment settings, like "Movie Night," create a tailored and immersive experience.

Additionally, this STEP provides flexibility to: Set custom access controls; Define restricted areas; Configure parental alerts. This integration between ekey and Control4 ensures every aspect of home automation is both personalized and secure.

Follow these STEPs to Program ekey bionyx Events:

1. Open Composer and navigate to the Programming Section.
2. In the project list, select the ekeyBoynxC4N driver.
3. From the event dropdown list, select one event at a time and program it as desired.



# Technical Support

## ekeybionyxC4N Driver Error List

Error code	Description
01	property label with nil vale
02	Action command in STEP 4 to generate user/device events was not confirmed. System events have not been updated. Try again and confirm yes when ready
03	Event deletion not confirmed. No changes made. Try again and confirm when ready.
04 (A, B)	Unable to decode bionyx JSON data file. Follow these STEPs:  Open "system_mapping_export.JSON" using Notepad text editor. Select and copy all the data in the file. In the Composer, go to the Action section and select the Command "STEP 1: Import bionyx." Paste the copied data into the 'Copy-Paste bionyx' field."
05	Unable to add the event. The event table might be corrupted or out of sync. Recommended actions: 1. Use the Action Command 'Delete ALL' and check the driver event list under the Programming section. - If the event list is empty, repeat STEPs 1 through 4. - If the event list is not empty, remove the driver from the project, re-add it, and repeat STEPs 1 through 4 as part of a new installation.
06	Invalid HTTP API message content or message part is out of sequence. This may indicate a communication error or a change in the message format from the bionyx message format.
07	Handle mismatch between expected message parts: HTTP part out of sequence or potential change in bionyx message format.
08	Access Token not configured in Driver Properties (Access Token). Please check and update.
09	Message rejected. Driver Access Token and bionyx Access Token do not match. Use the same token in Control4 Driver Properties (Access Token) and bionyx App Settings (Smart Home Connections > Configuration).
10	Message rejected. HTTPS message contains an invalid identifier, indicating a driver integrity issue
11	Message ignored: Invalid notification type received from bionyx. Valid input Types values: Finger=10, Digital Input=20
12 (A&B)	Message ignored: No matching finger or digital input event found, possibly due to a configuration error
13	Exceeded the maximum licensed user limit
14	Demo expired. This driver operates in demo mode for 5 days from the date of installation. After the demo period, a valid license is required to ensure continued functionality. To Purchase a License and Activate the Driver: Please contact us at <a href="mailto:license@ekeyusa.com">license@ekeyusa.com</a> and include the following information: <ul style="list-style-type: none"><li>○ Your Name</li><li>○ Phone Number</li><li>○ Billing Information</li><li>○ Control4 MAC Address</li></ul>

	<ul style="list-style-type: none"> <li>○ <b>Required Number of Users (20-100, in increments of 10)</b></li> <li>○ The basic system includes 2 I/O extensions per controller. Additional extensions require an IO Extension License (available for 2 to 18 extensions in increments of 2)</li> </ul>
15	Unable to decode JSON from the Finger notification. The message was ignored.
16	Exceeded the maximum licensed input-output extension limit per controller
17	DRM Controller ID not found in the Device List table. Possible mismatch with the Device table.
18	STEP 4: Generate eKey Bionyx Events not performed. Step 1 (Importing Bionyx Configuration) is missing. Complete Step 1 and try again.
19	STEP 2: Update DRM Controller I/O Extensions was not performed. Step 1 (Importing Bionyx Configuration) is missing. Complete Step 1 and try again

For technical support, please send an email to [info@ekeyUSA.com](mailto:info@ekeyUSA.com). Be sure to include and attach the following details:

- License key(s)
  - Control-4 controller details!
  - A brief description of the issue. Include lua output in the lua tab
  - Your contact information
  - Screenshots, files, and any other relevant information that might assist in resolving the issue.
- Providing this information will enable us to assist you most effectively and efficiently. Thank you.

## Change Log

Version	Release Date	Description
4	February 24, 2025	<ul style="list-style-type: none"><li>Separated licensing into User License and IO Extension License.</li><li>Enhanced recovery, error handling, and history logging.</li></ul>
3		Increase debug time and correction
2	January 21, 2025	Documentation correction and updated for improved clarity.
1	Dec 2024	Initial public availability of the driver.