

# **History Recorder**

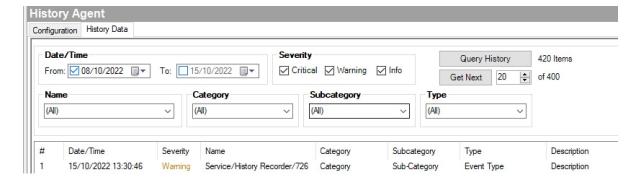
#### 1. Intro

Composer has the ability to save on History a custom event in the programming menu, but the info you can insert is only text pre-established and you can't use variables of the systems.

This driver allows you to save on History agent a custom record but also you can use any system information you want (variables). This is excellent if you want to troubleshoot something and record a variable in a determinate event, or if you just want to save a record of an important event to check it over the time.

#### **Features**

- The "Severity", "Category", "Subcategory", "Type" and "Description" content of the History table can be set by the driver.
- You can use system variables to insert on the fields
- History record action can be triggered with programming by an event.
- You can record the preset driver properties information, or you can custom change the history fields on the action part of the driver.



**Note:** This history events WON'T be available on the UI (the API doesn't have enough information about this, and this seems to be reserved for certain devices).

This is a free driver! The support is limited.

## 2. Properties

- Cloud Status: Current status of the driver license.
- Automatic Updates: Enable driver automatics updates (Recommended!)
- *Driver*: Driver name.
- Driver Version: Current driver version.
- Driver Information: Last debug information.
- Debug Mode: If On, driver will print debug information on Lua output.
- History Level: There are three levels to select: Critical, Warning, Info.
- History Category: Category field of History Agent. This property can have system variables.
- *History Sub-Category:* Subcategory field of History Agent. This property can have system variables.
- History Event Type: Event field of History Agent. This property can have system variables.
- *History Description:* Description field of History Agent. This property can have system variables.

### 3. Actions

• Test Record Preset: Save a history with the parameters of the properties.

#### 4. Commands

- \* Record History: Save a history with the parameters of the properties.
- \* Record Custom History: Save a history with the parameters below.

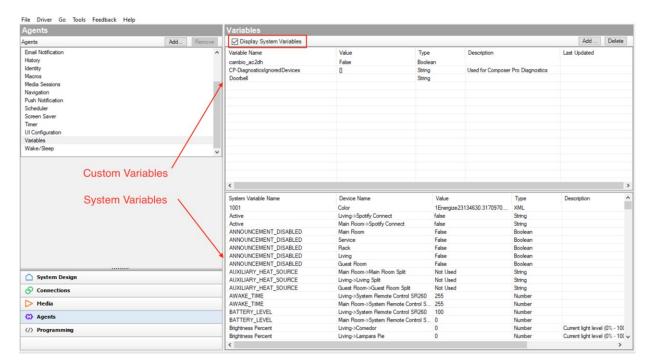
#### 5. Events

N/A

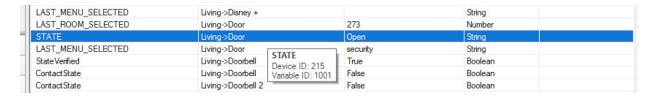
## 6. Using Real-Time variables

To use variables on the properties, we need the variables parameters. These are: deviceID and variableID.

To find them, we need to go to "Variables" Agent. If the variable is a custom variable (was created by the programmer), you will see it on the sheet. If the variable was created by the system (or another driver), you must check the box "Display System Variables".



Then, you must find the variable of interest, and then place the cursor above the variable to get the parameters:



In this case, deviceID=215 and variableID=1001.

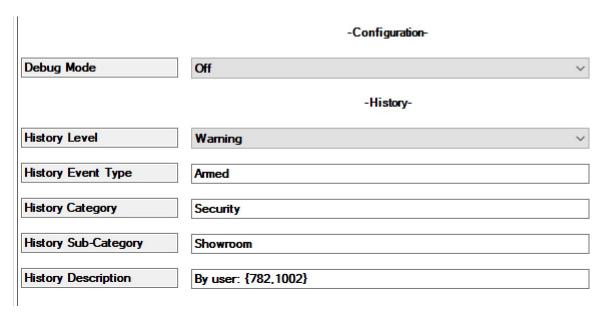
To use it on the properties, we must use this format: "{xxx,yyyy}" (without quotes) where "xxx" is the deviceID of the variable, and "yyyy" is the variableID.

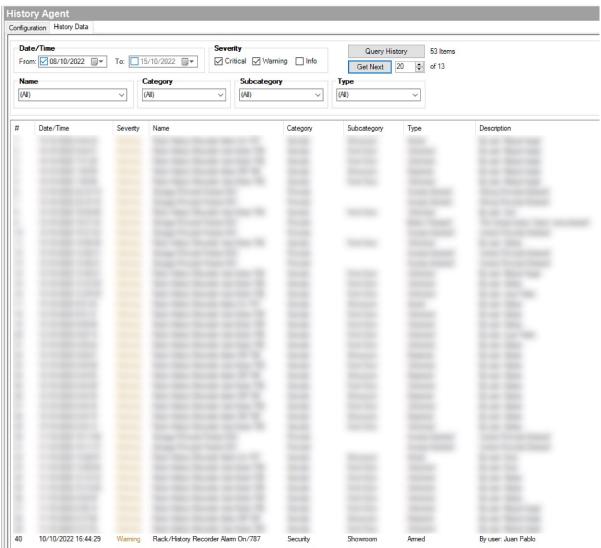
In this case would be "{215,1001}" (without quotes).

## 7. Example

Here you can see a working application of the driver. In this case we use a 2N RFID reader to trigger

to alarm the office when a valid TAG is held. When this happens, we trigger a history event to record who alarmed the office.





## 8. Warranty and disclaimer

**Barbini.dev** provide drivers tested as much as possible. However, due to the highly variants of systems setups and the constant changing of the electronics systems, updates and/or modifications may be required to fixing bugs or improve the driver functionality.

Many of our drivers interact with systems APIs (like Control4 and/or any other 3<sup>rd</sup> party system). Therefore, if any API changes, the driver may stop functionating correctly. **Barbini.dev** cannot guarantee long-term functionality of any driver developed that uses APIs. **Barbini.dev** has the right to repair, provide updates, or discontinue a driver at any time. These repairments or updates could be free or required additional expenses. Despite the above, our efforts will always be to maintain the drivers as long as possible and free of charge of upgrades as much as possible.

**Developed By: Barbini.dev** 

Contact: barbini.dev@gmail.com



-----

VERSION 22101506: Initial Release

VERSION 22101601: Added Record Custom History action