

BLUETOOTH

View online: <https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/bluetooth>

The **Bluetooth object** allows communicating with nearby Bluetooth devices via the [Web Bluetooth API](#).

You can find a simple example of using the Bluetooth object in the [Bluetooth - device name](#) example.

It's useful to have a basic understanding of the way Bluetooth works before starting to use the Bluetooth object. There is plenty of information covering the general technology if you search the web. In particular it is useful to know about services and characteristics. Familiarity with [Binary Data](#) will also be useful.

In Chrome, it can be useful to visit `chrome://bluetooth-internals` which provides a debug view of Bluetooth devices. This also helps identifying which services and characteristics a device exposes.

As of May 2022, the Bluetooth object is not supported in Android apps. This is because Google have not added support for the Web Bluetooth API in the Android WebView yet. (See [this issue](#) for updates). However it should work in the Chrome browser on Android.

Specifying services and characteristics

When specifying services and characteristics in the Bluetooth object, there are a few ways of identifying them:

- Use their standard e.g. `"heart_rate"` for the standard heart rate service. See the [standard service names](#) and [standard characteristic names](#). Note the prefixes are omitted, e.g. `org.bluetooth.characteristic.gap.device_name` is specified just as `gap.device_name` in the Bluetooth object.
- Use their 16-bit hexadecimal IDs e.g. `"0x1802"`. These can also be found in the linked standard lists, or vendor-specific IDs used.
- Use the full UUID, e.g. `"c48e6067-5295-48d3-8d5c-0395f61792b1"`. This is usually used to refer to a vendor specific characteristic or service.

Where a standard service is specified as a UUID, Construct will turn it back in to its standard name, e.g. `gap.device_name`.

Device IDs can also usually be omitted: if left empty, the first connected device will be used.

Bluetooth conditions

Is device connected

Test if a bluetooth device is currently connected by its device ID.

Is supported

True if bluetooth is supported on the current platform. The plugin will not work if this is false.

On device connected

On device disconnected

Triggered as devices connect (after *Request device* completes successfully) and disconnect.

The *DeviceID* expression is set to the ID of the relevant device.

On request device failed

Triggered after the *Request device* action if the user cancels or a Bluetooth connection was unable to be established.

On characteristic notification

On any characteristic notification

Triggered after starting notifications for a characteristic whenever a notification is received.

The notification data is loaded in to the provided Binary Data object. The *any* variant is triggered for all characteristics rather than a specific one, and sets the *CharacteristicID* expression accordingly

On characteristic read

On any characteristic read

On any characteristic read error

Triggered after the *Read value* or *Read binary* actions that attempt to read a characteristic value. If the read fails the error trigger fires. The *any* variant is triggered for any characteristic that successfully completes a read rather than a specific one, and sets the *CharacteristicID* expression accordingly. Once the read completes successfully, the characteristic value expressions are set, or if binary was read the data is now available in the chosen Binary Data object.

On characteristic written

On any characteristic written

On characteristic write error

Triggered after the *Write binary* action after the operation completes. If the write fails the error trigger fires. The *any* variant is triggered for any characteristic that successfully completes a write rather than a specific one, and sets the *CharacteristicID* expression accordingly.

On characteristics loaded

On characteristics error

Triggered after the *Request characteristics* action depending on the outcome. If successful the characteristic list is available with the *CharacteristicCount* and *CharacteristicAt* expressions.

On primary services loaded

On primary services error

Triggered after the *Request primary services* action depending on the outcome. If successful the primary service list is available with the *PrimaryServiceCount* and *PrimaryServiceAt* expressions.

Bluetooth actions

Request device

Prompt the user to choose a nearby Bluetooth device to connect to. This must be used in a user input trigger. The listed Bluetooth devices must be filtered by providing one of *Services* (or *Optional services*), *Device name* or *Device name prefix*; only matching devices will be shown. For possible ways to specify services, see *Specifying services and characteristics* above. *On device connected* triggers if successful, else *On request device failed*.

Disconnect device

Disconnect a connected Bluetooth device by its device ID.

Read value

Read binary

Read data from a Bluetooth characteristic. The device ID can be left blank to use the default device (the first device that is connected). For the service and characteristic, see *Specifying services and characteristics* above. One of the characteristic read triggers will run depending on the outcome. If successful, the selected Binary Data object will have the read data loaded for the *binary* variant, or the characteristic value expressions will be set for the *value* variant.

Write binary

Write data to a Bluetooth characteristic from a Binary Data object. The device ID can be left blank to use the default device (the first device that is connected). For the service and characteristic, see *Specifying services and characteristics* above. One of the characteristic write triggers will run depending on the outcome.

Start/stop notifications

Start or stop notifications for a Bluetooth characteristic. The device ID can be left blank to use the default device (the first device that is connected). For the service and characteristic, see *Specifying services and characteristics* above. While notifications are started, the notification triggers will run whenever a notification is received.

Request primary services

Request a list of primary services available for a device ID (which can be left blank to use the default device). If successful triggers *On primary services loaded*, and the primary service expressions can be used.

Browsers may filter the listed primary services depending on what was specified in Request device.

Request characteristics

Request a list of characteristics available for a service. If successful triggers *On characteristics loaded*, and the characteristic expressions can be used.

Bluetooth expressions

DeviceID

PrimaryServiceID

CharacteristicID

In a trigger, the ID of the relevant device, primary service, or characteristic, where applicable.

DeviceName

In *On device connected*, a string with the name of the device that connected.

CharacteristicSignedInteger

CharacteristicUnsignedInteger

CharacteristicFloat

CharacteristicText

After a successful *Read value* action, these expressions return the read value in a variety of different types. Construct will attempt to deduce the type from the length of the data, e.g. if the data is 1, 2 or 4 bytes long it will read it as both a signed and unsigned integer and set the expressions accordingly; if 4 or 8 bytes long set the float value; and always attempt to read the value as text. These values will only be meaningful if they match the type of value the characteristic really stores. For other data types, use the *Read binary* action instead.

PrimaryServiceCount

PrimaryServiceAt(index)

After *On primary services loaded*, use these expressions to retrieve the number of services and the service identifier at each index in the list.

CharacteristicCount

CharacteristicAt(index)

After *On characteristics loaded*, use these expressions to retrieve the number of characteristics and the characteristic identifier at each index in the list.