# DEVELOPER GUIDELINES ON SEMANTIC TAGS

## INTRODUCTION

A semantic tag is something that provides information to the openHAB User Interface about the class (i.e. the nature) of either things, or items linked to channels. Note the use of the word ‘class’.

Semantic tags are divided into the following categories: Equipment (relates to the class of a piece of equipment), Point (relates to the class of a particular data point within some equipment), Property (relates to the class of the information carried by a point), and Location (relates to the class of the location of an equipment or point).

## PURPOSE

The purpose of semantic tags is to help the openHAB User Interface generate an overview of the user’s home system with its respective nested rooms, devices (things) and items.

The purpose of this document is to provide rules for openHAB developers and openHAB maintainers in applying existing semantic tags, and in deciding whether proposed new semantic tags shall be permitted.

## GENERAL RULES

The various equipment, points, properties, and locations are displayed in the openHAB User Interface in a hierarchical tree structure, which is based on a hierarchy of semantic tags defined in openHAB Core. The list of semantic tags defined in openHAB Core is extensible. This document describes the developer rules for applying or extending this list.

Note: the core predefined semantic tags are presented to users as propositions, suggestions, or hints. It is possible for users to override or customise such tags at their own discretion. The remainder of this document does NOT relate to user overrides or customisation. It applies solely to the openHAB Core tags.

Rules for Applying the Core Predefined Tags:

1. Developers MUST apply core predefined tags in their addons where possible.
2. Developers MAY hard code the tags in the ‘thing-type.xml’ or apply them dynamically at run time.
3. Developers SHALL NOT hard code location tags in the ‘thing-type.xml’.
4. Developers MAY apply location tags dynamically at run time (if the addon can determine the location).
5. Developers SHALL NOT apply overridden or customised tags in their addons.
6. Developers MAY apply a point tag without a property tag (in cases where a property makes no sense).
7. Developers SHALL NOT apply a property tag without a point tag.
8. Semantic equipment tags MUST be applied at thing level.
9. Semantic point, property, and (subject to 4. above) location tags MUST be applied at channel level.
10. Developers MAY open a new PR to propose extensions to the core predefined tags as follows.

Rules for Extending the Core Predefined Tags:

1. New tags MUST fit within the hierarchy of semantic tags defined in openHAB Core.
2. New tags SHALL NOT be synonyms of existing tags e.g. kitchen/kitchenette, sitting/living room.
3. New tags MUST prove their targets cannot fit meaningfully under existing tags.
4. New tags MUST prove they would allow groups of three or more members.
5. New tags MUST prove they would apply to three or more bindings.
6. New tags SHALL NOT be single purpose (‘leaf’) tags.
7. Developers MUST state in their PR if the proposed tag is a ‘branch’ or subordinate (‘twig’) tag.
8. Developers MUST provide supporting evidence in their PR that the rules above and below are met.

## EQUIPMENT

An equipment tag describes the equipment class of a piece of equipment. Examples are light bulb, thermostat, pump, car, wall switch, push button, alarm system, or alarm detector.

1. New equipment tags MUST describe the equipment class of a physical or virtual device.
2. New equipment tags MUST be a noun.

## POINT

A point is a tag that describes the functional class (or purpose) of a data point within an equipment. Examples are measurement, control, set point, or status.

1. New point tags MUST describe the functional class (or purpose) of the data point.
2. New point tags MUST be what you do to, or how you get, the state of the Item.
3. New point ‘branch’ tags SHALL only very rarely be proposed.

## PROPERTY

A property is a tag that describes the information class (or content type) of the data produced or consumed by a point. Examples are temperature, brightness, on-off, time stamp

1. New property tags MUST describe the information class (or content type) of the point.
2. New property tags MUST be a noun and represent some attribute commonly controlled or sensed in a home automation context.
3. New property tags MUST be defined in conjunction with a functional class of a point tag.

## LOCATION

A location is a tag that describes the location class of a point. Examples are ground floor, dining (eating) room, kitchen (cooking room), living room, or garage.

1. New location tags SHALL NOT be synonyms (see rule 10.)
2. New location tags MUST describe the location class of the point.