

Jeedom OpenZwave Doc

Contents

1	INTRODUCTION	2
2	MANUFACTURER_SPECIFIC.XML	2
3	FICHER DE PARAMÈTRES	2
4	DÉFINITION DE L'ÉQUIPEMENT.....	3

1 Introduction

Le but de ce document est de fournir des informations sur la structure des fichiers de configuration des équipements OpenZwave dans Jeedom.

La définition d'un équipement OpenZwave dépend de différents fichiers.

2 Manufacturer_specific.xml

Ce fichier contient la liste de tous les équipements OpenZwave reconnu par Jeedom.

Emplacement:

.../plugins/openzwave/ressources/openzwave/config/manufacturer_specific.xml

Contenu:

Par constructeur, on retrouve une ligne par type de produit, son id, son nom et son fichier de configuration. Pour chaque type de produit, il peut y avoir plusieurs id. Ceci dépend des différentes zones de commercialisation des pays: EU, US, AU, ...

```
<Manufacturer id="010f" name="FIBARO System">
  <Product type="0501" id="0101" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0501" id="0102" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0501" id="0109" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0501" id="1002" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0501" id="3002" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0501" id="4002" name="FGBS001 Universal Binary Sensor" config="fibaro/fgbs001.xml" />
  <Product type="0100" id="0104" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0100" id="0106" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0100" id="0107" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0100" id="0109" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0100" id="100a" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0100" id="300a" name="FGD211 Universal Dimmer 500W" config="fibaro/fgd211.xml" />
  <Product type="0102" id="1000" name="FGD212 Dimmer 2" config="fibaro/fgd212.xml" />
  <Product type="0102" id="3000" name="FGD212 Dimmer 2" config="fibaro/fgd212.xml" />
  <Product type="0102" id="4000" name="FGD212 Dimmer 2" config="fibaro/fgd212.xml" />
</Manufacturer>
```

3 Fichier de paramètres

C'est le fichier de configuration repris dans Manufacturer_specific.xml pour l'équipement. Il contient la définition des paramètres de configuration de l'équipement.

Emplacement:

.../plugins/openzwave/ressources/openzwave/config/<<config>>

<<config>> est à remplacer par la valeur de config dans manufacturer_specific.

exemple:

Pour le module FGBS001 Universal Binary Sensor, config vaut fibaro/fgbs001.xml

L'emplacement complet du fichier sera donc:

.../plugins/openzwave/ressources/openzwave/config/fibaro/fgbs001.xml

Contenu:

Exemple:

Pour le FGBS001:

Le fichier n'est pas montré complet

```
<?xml version="1.0" encoding="utf-8"?>

<Product xmlns="http://code.google.com/p/open-zwave/">

  <!-- Configuration -->
  <CommandClass id="112">
    <Value type="short" genre="config" instance="1" index="1" label="IN1 Alarm Cancellation Delay" min="0" max="65535"
value="0" size="2">
      <Help>Input I alarm cancellation delay. Additional delay after an alarm from input IN1 has ceased. The parameter allows you
to specify additional time, after which the input no. 1 alarm is cancelled once its violation has ceased. Default 0.</Help>
    </Value>

    <Value type="short" genre="config" instance="1" index="2" label="IN2 Alarm Cancellation Delay" min="0" max="65535"
value="0" size="2">
      <Help>Input II alarm cancellation delay. Additional delay after an alarm from input IN1 has ceased. The parameter allows you
to specify additional time, after which the input no. 1 alarm is cancelled once its violation has ceased. Default 0.</Help>
    </Value>

    <Value type="list" genre="config" instance="1" index="3" label="Type of input no. 1" value="1" size="1">
      <Help>Type of input no. 1, what the input 1 will report if no contact is made. Default 1.</Help>
      <Item label="Input NO (Normal Open)" value="0" />
      <Item label="Input NC (Normal Close)" value="1" />
      <Item label="Input MONOSTABLE" value="2" />
      <Item label="Input BISTABLE" value="3" />
    </Value>

    <Value type="list" genre="config" instance="1" index="4" label="Type of input no. 2" value="1" size="1">
      <Help>Type of input no. 2, what the input 2 will report if no contact is made. Default 1.</Help>
      <Item label="Input NO (Normal Open)" value="0" />
      <Item label="Input NC (Normal Close)" value="1" />
      <Item label="Input MONOSTABLE" value="2" />
      <Item label="Input BISTABLE" value="3" />
    </Value>
    .... interrompu....
  </CommandClass>

  <CommandClass id="96" mapping="endpoints" />
  <CommandClass id="32" setasreport="true" />

  <!-- Association Groups -->
  <CommandClass id="133">
    <Associations num_groups="3">
      <Group index="1" max_associations="5" label="Input IN1" auto="false"/>
      <Group index="2" max_associations="5" label="Input IN2" />
      <Group index="3" max_associations="1" label="Lifeline" auto="true"/>
    </Associations>
  </CommandClass>

</Product>
```

4 Définition de l'équipement

On retrouve dans un directory les fichiers de configuration de l'équipement

Un fichier jpg qui contient l'icône de l'équipement

Un fichier json qui contient la configuration et les commandes

Emplacement:

../plugins/openzwave/core/config/devices/⟨⟨constructeur⟩⟩/⟨⟨equipement⟩⟩

⟨⟨constructeur⟩⟩ est construit de la manière suivante:

On utilise le nom du constructeur repris dans la config pour l'équipement donné.

Ici, c'est "fibaro"

On fait suivre ce nom de constructeur de la valeur du manufacturer id converti en decimal.

Ici, c'est 010f. donc en décimal: 271

⟨⟨constructeur⟩⟩ sera donc fibaro_271

⟨⟨equipement⟩⟩ est également construit sur des valeur extraites du fichier manufacturer_specific.xml

⟨⟨constructeur_id⟩⟩.⟨⟨product⟩⟩.⟨⟨id⟩⟩.⟨⟨name⟩⟩

⟨⟨constructeur_id⟩⟩ est la valeur du manufacturer id convertie en décimal.

010f ---> 271

⟨⟨product⟩⟩ est la valeur du product type convertie en décimal.

0501 ---> 1281

⟨⟨id⟩⟩ est la valeur de l'id convertie en décimal.

0101 ---> 257

⟨⟨name⟩⟩ est la valeur de name convertie en minuscule et les blancs remplacés par des points.

On arrive donc à ceci pour le premier équipement de l'exemple.

../plugins/openzwave/core/config/devices/fibaro_271/271.1281.257_fgbs001.universal.binary.sensor.json

Contenu:

Fichier jpg

C'est une image de l'équipement au format de 100x100 ou 100x85

fichier json

C'est le fichier qui va permettre la définition de l'équipement dans Jeedom.

Il contient plusieurs sections dont name, type et commands.

Name contient le nom de l'équipement

type contient le type de l'équipement; Par exemple Micromodule

Commands contient la définition de toutes les commandes qui vont être créées dans Jeedom

Exemple:

```
{
  "name": "FGBS-001 Universal Relay",
  "type": "Micromodule",
  ....
  "commands": [
    {
      "name": "Etat 1",
      "type": "info",
      "isVisible": 1,
      "isHistorized": 0,
      "configuration": {
        "class": 48,
        "value": "",
        "index": 0,
        "instance": 1
      },
      "subtype": "binary",
      "display": {
        "generic_type": "OPENING"
      },
      "template": {
        "dashboard": "door",
        "mobile": "door"
      }
    }
  ],
}
```

```
"name": "Etat 2",
"type": "info",
"isVisible": 1,
"isHistorized": 0,
"configuration": {
  "class": 48,
  "value": "",
  "index": 0,
  "instance": 2
},
"subtype": "binary",
"display": {
  "generic_type": "OPENING"
},
"template": {
  "dashboard": "door",
  "mobile": "door"
}
},
....
]
```