

# x0001 - feature set

Version 2

## FeatureSet

[0] **getCount()** → count

[1] **getFeatureID(featureIndex)** → featureID, featureType, featureVersion

## Overview

This feature allows the host to enumerate all the features present on a device.

## Functions and Events

### [0] **getCount()** → count

Returns the number of features contained in the set, **not including the root feature**.

#### Parameters

none

#### Returns

**count**

The number of features in the set, not including the root feature.

Table 1. *getCount()* response packet format

byte \ bit	7	6	5	4	3	2	1	0
0	count							
1-15	reserved							

### [1] **getFeatureID(featureIndex)** → featureID, featureType, featureVersion

Given a desired feature index, returns its featureID, the feature type bitfield, and the feature version.

#### Parameters

## featureIndex

The index in the feature table used to access the feature. Non-root features will start with index value 1.

Table 2. *getFeatureID() request packet format*

byte \ bit	7	6	5	4	3	2	1	0
0	featureIndex							
1-15	reserved							

## Returns

### featureID

The ID of the feature given by featureIndex.

### featureType

Bitfield describing the feature properties.

#### obsolete

An obsolete feature is a feature that has been replaced by a newer one, but is advertised in order for older SWs to still be able to support the feature (in case the old SW does not know yet the newer one).

#### hidden

A SW hidden feature is a feature that should not be known/managed/used by end user configuration SW. The host should ignore this type of features.

#### engineering

A hidden feature that has been disabled for user software. Used for internal testing and manufacturing.

#### manufacturing\_deactivatable

A manufacturing feature that can be permanently deactivated. It is usually also hidden and engineering.

#### compliance\_deactivatable

A compliance feature that can be permanently deactivated. It is usually also hidden and engineering.

## featureVersion

The zero based feature version number.

Table 3. *getFeatureID() response packet format*

byte \ bit	7	6	5	4	3	2	1	0
0	featureID (MSB)							
1	featureID (LSB)							

byte \ bit	7	6	5	4	3	2	1	0
2	featureType							
	obsl	hidden	eng	manuf_de act	compl_de act	---	---	---
3	featureVersion							
4-15	reserved							

#### NOTE

On feature versions: Features have a version number that starts at zero. When a feature version is incremented it means that new functionality has been added to it. Even if new functionality is present, newer versions of a feature will 'always' be backward compatible with all previous versions of a feature. Features that need to break backward compatibility will use a different feature number so de-facto creating a new feature.

## ChangeLog

- Version 2: Adds deactivatable manufacturing / compliance features bits in feature Type bitmap.
- Version 1: Adds the feature version return value (featVer) to feature function [\[1\]getFeatureID\(featIndex\)](#)
- Version 0: Initial version