



OPEN NETWORKING  
FOUNDATION

# Property errors Extension

*Version 0.1*

---

December 23, 2014



## Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Without limitation, ONF disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and ONF disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

No license, express or implied, by estoppel or otherwise, to any Open Networking Foundation or Open Networking Foundation member intellectual property rights is granted herein.

Except that a license is hereby granted by ONF to copy and reproduce this specification for internal use only.

Contact the Open Networking Foundation at <http://www.opennetworking.org> for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

WITHOUT LIMITING THE DISCLAIMER ABOVE, THIS SPECIFICATION OF THE OPEN NETWORKING FOUNDATION ("ONF") IS SUBJECT TO THE ROYALTY FREE, REASONABLE AND NONDISCRIMINATORY ("RANDZ") LICENSING COMMITMENTS OF THE MEMBERS OF ONF PURSUANT TO THE ONF INTELLECTUAL PROPERTY RIGHTS POLICY. ONF DOES NOT WARRANT THAT ALL NECESSARY CLAIMS OF PATENT WHICH MAY BE IMPLICATED BY THE IMPLEMENTATION OF THIS SPECIFICATION ARE OWNED OR LICENSABLE BY ONF'S MEMBERS AND THEREFORE SUBJECT TO THE RANDZ COMMITMENT OF THE MEMBERS.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>How it works</b>	<b>3</b>
<b>3</b>	<b>Property errors Experimenter ID</b>	<b>3</b>
<b>4</b>	<b>Property errors message</b>	<b>3</b>

## 1 Introduction

This document describes an ONF extension for OpenFlow version 1.3.X that returns errors for invalid properties in Table Feature set request or in some extension request (for example the ONF Bundle Extension for 1.3.X). When a standard error code is available, it should be preferred over those codes, for example some Table Feature errors are already supported in the `OFPET_TABLE_FEATURES_FAILED` type.

## 2 How it works

New error code are defined for malformed and incorrect properties.

## 3 Property errors Experimenter ID

The Experimenter ID of this extension is:

```
ONF_EXPERIMENTER_ID = 0x4F4E4600
```

## 4 Property errors message

The following errors are defined by this extension:

```
/* Error codes */
enum onf_prop_error_exp_type {
    ONFERR_ET_PROP_BAD_TYPE           = 4440, /* Unknown property type. */
    ONFERR_ET_PROP_BAD_LEN            = 4441, /* Length problem in property. */
    ONFERR_ET_PROP_BAD_VALUE          = 4442, /* Unsupported property value. */
    ONFERR_ET_PROP_TOO_MANY           = 4443, /* Can't handle this many properties. */
    ONFERR_ET_PROP_DUP_TYPE            = 4444, /* A property type was duplicated. */
    ONFERR_ET_PROP_BAD_EXPERIMENTER   = 4445, /* Unknown experimenter id. */
    ONFERR_ET_PROP_BAD_EXP_TYPE        = 4446, /* Unknown exp_type for exp-id. */
    ONFERR_ET_PROP_BAD_EXP_VALUE      = 4447, /* Unknown value for experimenter id. */
    ONFERR_ET_PROP_EPERM               = 4448, /* Permissions error. */
};
```

Those error codes use the following structure:

```
/* Message structure for all errors. */
struct onf_prop_error_msg {
    struct ofp_header header;
    uint16_t type;           /* OFPET_EXPERIMENTER. */
    uint16_t exp_type;      /* One of ONFERR_ET_PROP_* above. */
    uint32_t experimenter;  /* ONF_EXPERIMENTER_ID. */
    uint8_t data[0];        /* Up to 64 bytes of failed request. */
};
OFP_ASSERT(sizeof(struct onf_prop_error_msg) == sizeof(struct ofp_error_experimenter_msg));
```

The `type` field must be set to `OFPET_EXPERIMENTER`.

The `experimenter` field is the Experimenter ID (see 3).

The `data` fields contains a copy of the failed request message, truncated to 64 bytes.

The `exp_type` field is set to one of the error code defined above.