

User-Defined Errors for RSVP

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The Resource ReserVation Protocol (RSVP) defines an ERROR_SPEC object for communicating errors. That object has a defined format that permits the definition of 256 error codes. As RSVP has been developed and extended, the convention has been to be conservative in defining new error codes. Further, no provision for user-defined errors exists in RSVP.

This document defines a USER_ERROR_SPEC to be used in addition to the ERROR_SPEC to carry additional user information related to errors.

1. Introduction

The Resource ReserVation Protocol (RSVP) [RFC2205] defines an ERROR_SPEC object for communicating errors. That object has a defined format that permits the definition of 256 error codes. As RSVP has been developed and extended, the convention has been to be conservative in communicating errors. Further, no provision for user defined errors exists in RSVP.

When developing extensions to RSVP, it is often useful for those implementing to define error messages to aid both in the initial debugging and in testing against older versions or other implementations.

This document defines a new RSVP object to permit user-defined errors to be communicated. This will enable organizations to define errors that they can use for internal development. These error values could also be shared with the community at large to aid in promoting interoperability between diverse implementations.

RSVP PathErr and ResvErr messages require the presence of an ERROR_SPEC object ([RFC2205]). [RFC3473] defines the Notify message that also requires the presence of an ERROR_SPEC object. In order to not change the mandatory contents of these messages, this document defines a new error code value that indicates that the new object is present and carries a user-defined error code.

Note that the ResvConf message defined in [RFC2205] also carries an ERROR_SPEC object. But this usage of the object does not carry meaningful Error Codes or Error Values and so the extensions defined in this document are not applicable to that message.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. User-Defined Error

A new Error Code is defined for use in an ERROR_SPEC object.

Error Code = 33: User Error Spec

This error code is used to signal the presence of a USER_ERROR_SPEC. One Error Value is defined as follows.

Error Value 0 = Further details in User Error Spec

Further error values may be defined in future specifications.

When sending this error code, a USER_ERROR_SPEC object MUST be included in the PathErr, ResvErr, or Notify message.

3. USER_ERROR_SPEC Class

A new RSVP object class called USER_ERROR_SPEC is defined. To support backwards compatibility, its class number is in the range 192-247. As defined in [RFC2205], implementations that do not understand such an object will forward it unmodified.

USER_ERROR_SPEC object: Class = 194, C-Type = 1

Error Description

A string of characters padded with nulls (0x00) to a multiple of 4 bytes. According to the guidance in [RFC2277], this string MUST use UTF-8/Net-Unicode encoding [RFC5198]. Furthermore, it is RECOMMENDED that implementations limit the strings that they generate to single-line printable US-ASCII [ASCII] whenever feasible to improve the likelihood of easy use by the recipient.

If the Err Desc Len is zero, then no bytes are supplied.

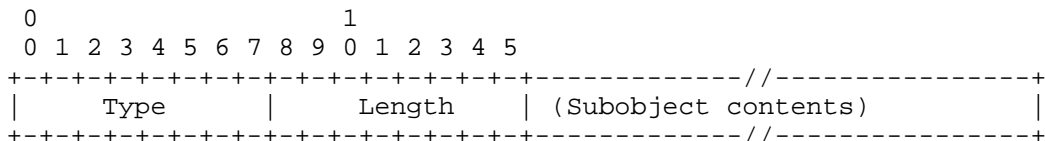
Note that the content of this field is implementation-specific. It is typically printable, but might not be shown to all users in all implementations (because of character set choice). Therefore, the content of the field SHOULD be limited to supplementary information and SHOULD NOT contain information critical to operating the network. Critical information is present in the User Error Value field.

User-Defined Subobjects

User-defined subobjects MAY be included. The generic format of subobjects is specified in Section 3.1. The semantics of a subobject is indicated by the Type field, but the semantics, format, and contents of the Value field are specified by the (sub-)organization indicated by the Enterprise Number and Sub Org fields of this object.

3.1. Subobjects

Each subobject is encoded as a TLV in the following format:



Type

An 8-bit number assigned by the (sub-)organization indicated by the Enterprise Number and Sub Org fields of the USER_ERROR_SPEC object.

Length

The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length MUST be at least 4, and MUST be a multiple of 4.

4. Procedures for Using the User Error Spec

4.1. Procedures for Sending the User Error Spec

A `USER_ERROR_SPEC` object MAY be included in any `PathErr`, `ResvErr`, or `Notify` message for any defined error code. The Enterprise Number MUST be a valid value assigned by IANA from the "Enterprise Numbers" registry.

As specified in [RFC2205] and [RFC3473], an `ERROR_SPEC` object with a valid error code MUST be included in all `PathErr`, `ResvErr`, and `Notify` messages. This rule is not changed by these procedures even when a `USER_ERROR_SPEC` object is included. If no other error code applies, the Error Code in the `ERROR_SPEC` object MUST be set to "User Error Spec" as defined in Section 2 of this document. When the Error Code in the `ERROR_SPEC` object is set to "User Error Spec", the Error Value sub-code SHOULD be set to "Further details in User Error Spec" as defined in Section 2, but further Error Value sub-codes may be defined in future specifications.

4.2. Procedures for Receiving the User Error Spec

It is RECOMMENDED that implementations that receive a `PathErr`, `ResvErr`, or `Notify` message carrying a `USER_ERROR_SPEC` object log (at a minimum) the Enterprise Number, Sub-organization, User Error Value, and Error Description. Note that the character set used for the Error Description may mean that it might not be suitable for display of logging in all systems. Implementations capable of interpreting the contents of the `USER_ERROR_SPEC` object SHOULD take further action based on the reported error.

Implementations that are not UTF-8 capable and that receive a `USER_ERROR_SPEC` object SHOULD handle the Error Description according to the procedures set out in [RFC5137].

If a message is received containing an `ERROR_SPEC` object using the "User Error Spec" error code, but not containing a `USER_ERROR_SPEC` object, the message MUST be treated as malformed and handled according to [RFC2205].

Implementations SHOULD ignore repeated occurrences of USER_ERROR_SPEC objects, and SHOULD forward them unchanged on any messages that they forward. This provides for forward compatibility.

Implementations receiving a USER_ERROR_SPEC object on some message other than a PathErr, ResvErr, or Notify message MUST treat the error as a malformed message and process according to [RFC2205].

5. IANA Considerations

5.1. RSVP Error Codes

This document makes the following assignments from the RSVP Error Codes and Globally-Defined Error Value Sub-Codes registry:

Error Code	Meaning
33	User Error Spec

One Error Value sub-code is defined for use with this Error Code as follows:

0 = Further details in User Error Spec

5.2. RSVP Objects

This document makes the following assignments from the RSVP Class Names, Class Numbers, and Class Types registry:

Number Space	Value	Name
Class Numbers	194	USER_ERROR_SPEC
Class Type	1	User-Defined Error

6. Security Considerations

This document makes no changes to the basic message exchanges of [RFC2205] and [RFC3473]. It will result in a small increase in the number of error messages sent in cases where messages were previously silently dropped due to the lack of an appropriate error code.

The mechanisms defined in this document may be used by implementations to report additional error conditions and information arising from security issues and attacks on the RSVP network.

Note that the use of a character string that will be displayed or logged opens the potential for certain security attacks through the use of overruns or embedded control characters. Implementations SHOULD take precautions against overruns, and (especially where the full character set of [RFC5198] is not supported, SHOULD use the procedures set out in [RFC5137] to handle unexpected or unknown control characters.

7. Acknowledgments

The authors would like to thank Elisheva Halevy for motivating this document. Thanks to Tom Nadeau, Magnus Westerlund, Hannes Tschofenig, Bruce Davie, Jukka Manner, Francois Le Faucheur, Lars Eggert, and Tom Petch for their review and comments.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC5137] Klensin, J., "ASCII Escaping of Unicode Characters", BCP 137, RFC 5137, February 2008.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.
- [ASCII] American National Standards Institute, "USA Code for Information Interchange", ANSI X3.4, 1968.

8.2. Informative References

- [RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

Authors' Addresses

George Swallow
Cisco Systems, Inc.
EMail: swallow@cisco.com

Adrian Farrel
Old Dog Consulting
EMail: adrian@olddog.co.uk

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.