Advanced Exception Information Format (AXIF)
draft-luis140219-axif-04

Abstract

This document defines the Advanced Exception Information Format (AXIF), an open and extensible format for providing information about errors and exceptions that allows transmission of information about multiple exceptions and the relations between them.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 4, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ......................................................... 2
2. The format .......................................................... 2
3. Values for child OIDs of 1.3.6.1.4.1.37476.9000.46.2.19 ........ 2
   3.1. NTSTATUS ...................................................... 2
   3.2. Level .......................................................... 3
This document specifies the Advanced Exception Information Format (also known as AXIF), a format defined using ASN.1 [X.680].

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].

The OID 1.3.6.1.4.1.37476.9000.46.2.19 is allocated to this format.

The format is the application of the Basic Encoding Rules [X.690] to instances of the AXIF type in the ASN.1 module in Appendix A. As a consequence of the choice of encoding rules, AXIF is a binary file format.

Each ExceptionInformation in an AXIF file/transmission is a list of data structures that contain a property OID, the referencesOthers boolean (Section 4), and a value for the property identified by the OID. Rules for values of properties with child OIDs of 1.3.6.1.4.1.37476.9000.46.2.19 are defined in Section 3.

Unless explicitly stated otherwise for a particular OID, for OIDs in this section that are AXIF properties, the referencesOthers boolean for instances of properties identified by the OID MUST be false.
3.1. NTSTATUS

This type is used in an ExceptionInformation to indicate the NTSTATUS value (Section 2.3.1 of [MS-ERREF]) of the exception corresponding to the ExceptionInformation.

The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.1.

3.2. Level

This type is used in an ExceptionInformation to indicate the error level of the exception: if it was a direct problem, other problem that involved clients or just other, internal problem. A value of this type is of the ASN.1 type Level, defined in Appendix A.

The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.2.

3.3. Basic Information

This type is used in an ExceptionInformation to indicate information about the system in which the exception corresponding to the ExceptionInformation occurred. A value of this type is of the ASN.1 type BasicInformation, defined in Appendix A.

The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.3.

3.4. Reserved

OID 1.3.6.1.4.1.37476.900.46.2.19.4 is reserved for identifying the ASN.1 module in Appendix A.

This section is just here for consistency in the numbering.

3.5. x86 Context

This type is used in an ExceptionInformation to indicate the context in which an exception occurred in a x86 system. A value of this type is encoded in ASN.1 as an OCTET STRING with exactly 204, 716 or 720 octets, using the CONTEXT C structure as defined in WinNT.h when targeting x86, being all fields in little-endian. If the system is not x86, or the process that generated the exception is not a x86 32-bit process in a 64-bit operating system, this type MUST NOT be used. With exactly 204 octets, the ExtendedRegisters field is removed from the structure and all attempts to extract data from it MUST fail. The ASN.1 type for encoding values of this type is called X86Context in the ASN.1 module in Appendix A.

The structure’s definition can be obtained at [X86CTX].
The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.5.

3.6. Win32 Exception Record

This type is used in an ExceptionInformation to indicate an exception record describing the exception. A value of this type is of the ASN.1 type ExceptionRecord, defined in Appendix A. This ASN.1 type is an 80-octet OCTET STRING representing an EXCEPTION_RECORD C structure, being all fields in little-endian, and is called Win32ExceptionRecord in the ASN.1 module of Appendix A.

The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.6.

3.7. Memory Map

This type is used in an ExceptionInformation to indicate a total or partial memory map of the process in which the exception occurred. A value of this type is of ASN.1 type MemoryMap, defined in Appendix A.

The protection value is an integer working as a bitfield, for which the following flags are defined [MEMINFO, MEMPROTC]:
* 0x00000001 - PAGE_NOACCESS
* 0x00000002 - PAGE_READONLY
* 0x00000004 - PAGE_READWRITE
* 0x00000008 - PAGE_WRITECOPY
* 0x00000010 - PAGE_EXECUTE
* 0x00000020 - PAGE_EXECUTE_READ
* 0x00000040 - PAGE_EXECUTE_READWRITE
* 0x00000080 - PAGE_EXECUTE_WRITECOPY
* 0x00000100 - PAGE_GUARD
* 0x00000200 - PAGE_NOCACHE
* 0x00000400 - PAGE_WRITECOMBINE
* 0x00000100 - MEM_COMMIT
* 0x00000200 - MEM_RESERVE
* 0x00010000 - MEM_FREE
* 0x00020000 - MEM_PRIVATE
* 0x00040000 - MEM_MAPPED
* 0x01000000 - MEM_IMAGE
* 0x40000000 - PAGE_TARGETS_NO_UPDATE

A memory map, if its total boolean is true, MUST contain information about all blocks of memory that is allocated at the time of the exception in any way (including, but not limited to, stack, heap, code, data and mapped files).

For the flags, the rules specified in [MEMINFO] and [MEMPROTC] for the bits each one defines MUST be met.
The OID for specifying a value of this type is 1.3.6.1.4.1.37476.9000.46.2.19.7.

4. The referencesOthers boolean

The referencesOthers boolean in an ExceptionInformationRecord MUST properly indicate whether it references other ExceptionInformationRecord structures in the same ASN.1 transmission by using the indices of those records within the transmission.

5. Security Considerations

Implementations MUST assure server safety in the event of exceptions and send valid structures according to the ASN.1 Basic Encoding Rules [X.690].

6. IANA Considerations

A new MIME type for AXIF may need to be registered.

7. References

7.1. Normative References


7.2. Informative References

[X86CTX] NirSoft, "struct CONTEXT",
https://www.nirsoft.net/kernel_struct/vista/CONTEXT.html

Appendix A. ASN.1 Module

This module is written in 2002 ASN.1 [X.680].

AXIF { iso(1) identified-organization(3) dod(6) internet(1) private(4)
   enterprises(1) 37476 freeoid(9000) 46 standard(2) axif(19) module(4) }

DEFINITIONS IMPLICIT TAGS ::= 

BEGIN

-- EXPORTS ALL --

IMPORTS AlgorithmIdentifier, Certificate FROM PKIX1Explicit88 { iso(1)
   identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit(18) }

id-axif OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6)
   internet(1) private(4) enterprises(1) 37476 freeoid(9000) 46
   standard(2) axif(19) }

AXIF ::= SEQUENCE {
   info SEQUENCE OF ExceptionInformation,
   signature DigitalSignature
}

DigitalSignature ::= SEQUENCE {
   signatureAlgorithm AlgorithmIdentifier,
   -- first certificate MUST sign the message
   certificates [0] SEQUENCE OF Certificate,
   signature BIT STRING
}

ExceptionInformation ::= SEQUENCE OF ExceptionInformationRecord

ExceptionInformationRecord ::= SEQUENCE {
   type OBJECT IDENTIFIER,
   referencesOthers BOOLEAN DEFAULT FALSE
   value ANY DEFINED BY type
}

-- Specifying a NTSTATUS value in an exception information record

id-axif-ntstatus OBJECT IDENTIFIER ::= { id-axif ntstatus(1) }

NTSTATUS ::= INTEGER (-2147483648..2147483647)
-- Specifying a level in an exception information record

id-axif-level OBJECT IDENTIFIER ::= { id-axif level(2) }

-- internal is for internal issues
-- involvesClients is for errors that involve clients (for instance,
-- sharing violations between a client and a server)
-- direct is in the case a file the user requests was not found in the
-- server
AXIFLevel ::= ENUMERATED { internal, involvesClients, direct }

-- Specifying basic information in an exception information record

id-axif-basicInformation OBJECT IDENTIFIER ::= { id-axif
basicInformation(3) }

BasicInformation ::= SEQUENCE {
  client BOOLEAN DEFAULT FALSE,
  data String
}

String ::= CHOICE {
  ia5String IA5String,
  utf8String UTF8String,
  printableString PrintableString,
  bmpString BMPString
}

-- Specifying a x86 context in an exception information record

id-axif-x86Context OBJECT IDENTIFIER ::= { id-axif x86Context(5) }

X86Context ::= OCTET STRING (SIZE(204 | 716 | 720))

-- Specifying a Win32 exception record in an exception information
-- record

id-axif-win32ExceptionRecord OBJECT IDENTIFIER ::= { id-axif
win32ExceptionRecord(6) }

Win32ExceptionRecord ::= OCTET STRING (SIZE(80))

-- Specifying a memory map in an exception information record

id-axif-memoryMap OBJECT IDENTIFIER ::= { id-axif memoryMap(7) }

MemoryMap ::= CHOICE {
  partialMap SET OF MemoryMapEntry,
  totalMap [0] SET OF MemoryMapEntry
}
MemoryMapEntry ::= SEQUENCE {
    address INTEGER(0..MAX),
    size INTEGER(1..MAX),
    protection INTEGER,
    filepath [0] EXPLICIT String OPTIONAL,
    type [1] MemoryType OPTIONAL
}

MemoryType ::= ENUMERATED { stack(0) heap(1) managed-stack(2)
    managed-heap(3) other-stack(4) other-heap(5) }

END

Author’s Address

Luis Camara (@luis140219 on GitHub)
Praceta das Tilias 102 R/C A
S. Pedro do Estoril
2775-336 Cascais
Portugal
Phone: +351 91 414 09 04
EMail: luis.camara@live.com.pt