

**Java IDX Format Specification**

# Introduction

A Java IDX file is used by the Java runtime to store details about Java files that have been downloaded. This specification was generated by source code analysis and reverse engineering

This document is intended as a working document for the Java IDX format, which should allow existing Open Source forensic tooling to be able to process this file type.

License

Copyright (c) 2013 Mark Woan<markwoan@gmail.com>.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Format

Currently there appears to be three different formats used by the Java runtime:

* 6.0.2
* 6.0.3/6.0.4
* 6.0.5

The header (Section 1) is a fixed at 128 bytes. The file format is Big Endian. All dates are stored as Java Epoch’s e.g. MS from 1/1/1970 00:00:00

Version 6.0.5

**Section 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 1 |  | Is busy flag |
| 1 | 1 |  | Incomplete flag |
| 2 | 4 | 605 (0x25D) | Cache version |
| 6 | 1 |  | Is shortcut image flag |
| 7 | 4 |  | Content length |
| 11 | 8 |  | Last modified |
| 19 | 8 |  | Expiration date |
| 27 | 8 |  | Validation timestamp |
| 35 | 1 |  | Known to be signed flag |
| 36 | 4 |  | Section 2 length |
| 40 | 4 |  | Section 3 length |
| 44 | 4 |  | Section 4 length |
| 48 | 4 |  | Section 5 length |
| 52 | 8 |  | Blacklist validation time |
| 60 | 8 |  | Cert expiration date |
| 68 | 1 |  | Class verification Status |
| 69 | 4 |  | Reduced manifest length |
| 73 | 4 |  | Section 4 pre15 length |
| 77 | 1 |  | Has only signed entries flag |
| 78 | 1 |  | Has single code source flag |
| 79 | 4 |  | Section 4 certs length |
| 83 | 4 |  | Section 4 signers length |
| 87 | 1 |  | Has missing signed entries flag |
| 88 | 8 |  | Trusted libraries validation time |
| 96 | 4 |  | Reduced manifest 2 length |
| 100 | 1 |  | Is proxied flag |

**Section 2**

If the section 2 length is greater than 0 then the section 2 can be processed by seeking to offset 128 (0x80). The strings within this section are prefixed with the string lengths and are null terminated.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 2 |  | Version string length |
| 2 |  |  | Version string |
|  | 2 |  | URL string length |
|  |  |  | URL string |
|  | 2 |  | Namespace ID string length |
|  |  |  | Namespace string |
|  | 2 |  | Codebase IP string length |
|  |  |  | Codebase IP string |

The response HTTP headers are stored directly after the section 2 header. The headers can be processed in a loop, repeating until the count of headers matches the value parsed from the first offset. If the header name string equals “<null>”, then the header name string should be cleared, and left blank as it is used to store the HTTP response code and description.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 4 |  | Count of HTTP headers |
| 4 | 2 |  | Header name length |
|  |  |  | Header name string |
|  | 2 |  | Header value length |
|  |  |  | Header value string |

Version 6.0.3/6.0.4

**Section 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 1 |  | Is busy flag |
| 1 | 1 |  | Incomplete flag |
| 2 | 4 | 603 (0x025B) /604 (0x025C) | Cache version |
| 6 | 1 |  | Not used |
| 7 | 1 |  | Not used |
| 8 | 1 |  | Is shortcut image flag |
| 9 | 4 |  | Content length |
| 13 | 8 |  | Last modified |
| 21 | 8 |  | Expiration date |
| 29 | 8 |  | Validation timestamp |
| 37 | 1 |  | Known to be signed flag |
| 38 | 4 |  | Section 2 length |
| 42 | 4 |  | Section 3 length |
| 46 | 4 |  | Section 4 length |
| 50 | 4 |  | Section 5 length |
| 54 | 8 |  | Blacklist validation time |
| 62 | 8 |  | Cert expiration date |
| 70 | 1 |  | Class verification Status |
| 71 | 4 |  | Reduced manifest length |
| 75 | 4 |  | Section 4 pre15 length |
| 79 | 1 |  | Has only signed entries flag |
| 80 | 1 |  | Has single code source flag |
| 81 | 4 |  | Section 4 certs length |
| 85 | 4 |  | Section 4 signers length |
| 89 | 1 |  | Has missing signed entries flag |
| 90 | 8 |  | Trusted libraries validation time |
| 98 | 4 |  | Reduced manifest 2 length |

**Section 2**

See the Section 2 format for version 6.0.5

Version 6.0.2

**Section 1**

The strings within this section are prefixed with the string lengths and are null terminated.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 1 |  | Is busy flag |
| 1 | 1 |  | Incomplete flag |
| 2 | 4 | 602 (0x025A) | Cache version |
| 6 | 1 |  | Not used |
| 7 | 1 |  | Not used |
| 8 | 1 |  | Is shortcut image flag |
| 9 | 4 |  | Content length |
| 13 | 8 |  | Last modified |
| 21 | 8 |  | Expiration date |
| 29 | 8 |  | Validation timestamp |
| 37 | 2 |  | Version string length |
|  |  |  | Version string |
|  | 2 |  | URL string length |
|  |  |  | URL string |
|  | 2 |  | Namespace ID string length |
|  |  |  | Namespace ID string |

The response HTTP headers are stored directly after the section 2 header. The headers can be processed in a loop, repeating until the count of headers matches the value parsed from the first offset. If the header name string equals “deploy\_resource\_codebase\_ip”, then the Codebase IP property/attribute can be set to the “header” value.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Value** | **Description** |
| 0 | 4 |  | Count of HTTP headers |
| 4 | 2 |  | Header name length |
|  |  |  | Header name string |
|  | 2 |  | Header value length |
|  |  |  | Header value string |

History

**v1.0.0**

* Initial release