

**ECMA**

Standardizing Information and Communication Systems

---

**Common Language Infrastructure (CLI)**

**Technical Report:**

**Information Derived from Partition IV XML File**



This Technical Report is intended as an aid for understanding the libraries specified in Standard ECMA-335, Partition IV: Profiles and Libraries. That Partition includes a machine-readable specification, in XML, of the types that comprise the standard libraries. This Technical Report, in companion files, provides the following items which help to form a traceable chain from the normative XML specification to a portable, printable representation of its contents:

1. **Tool Source Code.** A program written in the C# programming language, XML Style-sheet Language (XSL), and using the facilities of the Microsoft .NET Framework™ and Microsoft Office™ to convert the XML into files viewable using Microsoft Word™. This program, provided by Intel Corporation, can be modified to produce other views of the XML.
2. **Microsoft Word™ Files.** These are the files produced by running the tool mentioned above on the XML from Partition IV. The ECMA task group TC39/TG3 used similar files (produced using earlier versions of this tool run against earlier versions of the XML) as the primary means of reviewing the XML.
3. **PDF™ Files.** These files are produced from the Microsoft Word™ files using the Adobe Acrobat™ program. They are viewable on a wide range of computer systems and printable on a range of computer output devices. In most cases, they will appear visually identical regardless of the means used to render them.

Partition IV normatively specifies the format of the XML file. The tool provided here renders all parts of the XML with the following exceptions, which may be inspected directly in the XML file:

- Name (this is redundant and unnecessary since the FullName of the type is rendered)
- FullNameSP (this is redundant and unnecessary since the FullName of the type is rendered)
- AssemblyInfo.AssemblyCulture (this is reserved for future use; currently its value is 0)
- AssemblyInfo.Attributes.Attribute.Excluded (if its value is 0, it is not rendered, but if it is 1, the library that is necessary for inclusion is listed)
- TypeExcluded (as above)
- Interfaces.Interface.Excluded (as above)
- Attributes.Attribute.Excluded (as above)
- Attributes.Attribute.ExcludedTypeName (the short Name is rendered, however)
- Members.Member.ReturnValue.ReturnType (the Fullname of the type of the return value is not specified, but it is implied via the member's signature)
- Members.Member.Parameters.Parameter.Type (the Fullname of the type of the parameter is not specified, but it is implied via the member's signature)
- The "value\_" field for enums.
- Member.Member.Docs.altcompliant (used on methods/properties that have a CLSCompliant(false) attribute tag, and specifies a CLS-compliant method/property that can be use as an alternative. The remarks/description section usually specifies this anyway).
- Member.Member.Docs.altmember (used on methods/properties that have equivalent alternatives that may be used. For example, the string op\_equality operator has the altmember element and specifies the String.Equals() method as an equivalent alternative. The remarks/description section usually specifies this anyway).