

Appearance Exchange Format (AxF)

Frequently Asked Questions

Date: 3/29/2018



- [Backward Compatibility: Can a new AxF library read a file or data within a file created by an older library?](#)
- [Forward Compatibility: Can an older AxF library read a file or data within a file created by a new library?](#)
- [Exists there an option to downgrade an existing AxF file if my application/library does not support the features it contains?](#)

[Backward Compatibility: Can a new AxF library read a file or data within a file created by an older library?](#)

Any given AxF library version is designed to read all existing AxF files from that or any prior release.

This is assured (1) by using HDF5 as binary base layer of AxF, which makes an identical backward compatibility statement with respect to HDF5 releases (see [HDF5 documentation](#)), and (2) by storing versioning information along with AxF entities. This includes the overall version number of the AxF file format as well as specific version numbers for each material representation.

[Forward/Upward Compatibility: Can an older AxF library read a file or data within a file created by a newer library?](#)

Any given AxF library version can additionally read AxF files written by future releases with the same major version (i.e. currently AxF 1.x), provided that the AxF file contains at least one supported material representation.

This is assured by the *maximum compatibility* principle, according to which the AxF library writes AxF files: Rather than always using the latest version possible (which would prevent forward compatibility), the AxF library writes a given material representation in the minimum representation version that supports all required features. Then, the file format version is chosen based on the minimum version of all material representations in the file. If necessary, the file format is automatically downgraded to that version.

This should provide best forward compatibility by allowing a certain new AxF file to be read by a maximum number of older AxF libraries/applications, depending on which AxF features have been used when creating the AxF file.

[Exists there an option to downgrade an existing AxF file if my application/library does not support the features it contains?](#)

Since the AxF file format is already automatically downgraded to the lowest version that is possible without loss of information (see above), there is no way to manually downgrade existing AxF files.

In general, it is advisable to consider the compatibility of an AxF file already during creation. X-Rite's Pantora application for example informs the user about the AxF version number associated with a given feature and also allows to enforce compatibility to AxF version 1.0 (baseline).

Additionally, it is possible to store multiple representations of a material with different AxF versions in a single AxF file. This allows to achieve maximum forward compatibility on the hand, and at the same time also the highest possible quality and expressiveness provided by a recent AxF version.