

[How to determine the correctness of an ONVIF implementation? Can test tool alone be the judgement of such correctness?](#)

[What is ONVIF spec set, and why spec set has versions?](#)

[If ONVIF testing result in compatibility issues, should one modify camera firmware or NVR software to achieve compatibility?](#)

[What does ONVIF version 1.0, 1.1, 1.2, 2.0 generally means? Why version 1.0, 1.1, 1.2, 2.0 can be incompatible?](#)

[What is XML namespace?](#)

[As far as I understand, ONVIF only supports video streaming, right? Other functions, such as changing resolution, frame-rate, quality, brightness, motion detection, input signal, output relays, or PTZ aren't defined in ONVIF.](#)

[Which version of ONVIF specification does Genius Vision support? 1.0, 1.1, 1.2, or 2.0?](#)

[I have some questions about "ONVIF verification program" in regards to service terms or business matters, how can I get answers?](#)

How to determine the correctness of an ONVIF implementation? Can test tool alone be the judgement of such correctness?

The correctness of an ONVIF implementation is multi-aspect and it cannot be judged alone by test tool because (1) test tool can contain bugs(2) test tool does not cover all testing scenarios. This is the root cause of the industry reality that many ONVIF implementations doesn't work together well, but they can all pass test tool.

According to the ONVIF testing specification document "ONVIF_Test_Specification_v13_06.pdf", in "4.Scope" it says "This ONVIF Test Specification does not cover the complete set of functions as defined in [ONVIF Network Interface Specs]; **instead it covers a subset of it**". This simply says that the official testing specification does not cover all scenarios and therefore is incomplete. In the same document, at "5. Requirements for respective test cases", it says "This document defines whole requirements for claiming conformance to the ONVIF standard which includes every aspect of each requirement in the above documents. ". This means under the official standard, one can claim ONVIF conformance by merely passing those testing specification. Using deductive logic by combining the two points, one can get the conclusion that "claiming ONVIF conformance does not automatically guarantee the completeness of an ONVIF implementation".

This does not say that ONVIF official testing specification lacks completeness, but merely pointing out a fact that there can hardly be any automated testing process that can achieve true completeness. If one really wants to do "complete testing", it might only cost too much beyond reasonable economic cause. In contrast, if we take a step back, you'll see that users do not demand real "completeness", they just want components to work together in a reasonable manner, because in real world, demanding 100% of anything is almost not possible all the time.

Thus, one must take "**human knowledge**" into consideration to determine what device or technology is "usable & mature", and not rely such determination on an automated testing tool.

The best proof of the reasoning above is that the test tool cannot determine if the video image correctly reflects the user configuration change. For example, if we have a test specification like the following one:

- Change the configuration of a camera in resolution, frame-rate, bit-rate, brightness, saturation, and hue, then check to see if the camera output video indeed conforms the requested configuration change.

You will find that this test specification is **not** covered by the ONVIF test tool. Why is such an obviously essential test case, as this example, not covered by the testing tool? Educated guess is that since the only official way to verify ONVIF conformance is to use test tool, then it means that all testing specification must be able to be programmed and verified using the testing tool. However, "to determine if the camera video output conforms to a configuration change" seems to fall beyond the comfort range of current computer programming technology (meaning it could be done, but might come with extreme cost that render it impractical). In conclusion, using the technology readily available today, human intervention to ONVIF verification is still the best and most practical way. Nonetheless, human intervention inevitably introduces subjective factors, so it's not easy to write down a specification for that.

In conclusion, the correctness of an ONVIF implementation is judged by reading the ONVIF specification (not testing specification). ONVIF specification is a "set of documents", and together they form a "[spec set with version](#)". Both sides of integration (NVC and NVT) shall use ONVIF specification set to communicate standard conformance issues. Currently Genius Vision implements ONVIF using ONVIF spec set version 2.3. Please note this "version 2.3" is not entirely the same meaning as the commonly referred "[ONVIF 1.0, 1.2, 2.0](#)".

What is ONVIF spec set, and why spec set has versions?

ONVIF specification is composed by multiple specifications, therefore it's called "spec set". For accuracy, in addition to each version of each spec, ONVIF defines a group of a specific version of specification and assign the set a "spec version". For example, spec set 2.3 is composed of following specifications (and their versions):

- ONVIF-AccessControl-Service-Spec-v100.pdf
- ONVIF-ActionEngine-Service-Spec-v100.pdf
- ONVIF-DeviceIo-Service-Spec-v221.pdf
- ONVIF-Display-Service-Spec-v230.pdf
- ONVIF-DoorControl-Service-Spec-v100.pdf
- ONVIF-Imaging-Service-Spec-v221.pdf
- ONVIF-Media-Service-Spec-v230.pdf
- ONVIF-PTZ-Service-Spec-v221.pdf
- ONVIF-Receiver-Service-Spec-v221.pdf

If ONVIF testing result in compatibility issues, should one modify camera firmware or NVR software to achieve compatibility?

Such issues should be resolved by further discussing ONVIF specification to determine which part needs modification, camera firmware or NVR software.

Generally speaking, if the compatibility issue under discussion falls in the gray area of specification, we will modify NVR software to add flexibility (such as adding options), but it's not a hard rule.

What does ONVIF version 1.0, 1.1, 1.2, 2.0 generally means? Why version 1.0, 1.1, 1.2, 2.0 can be incompatible?

The design principle of ONVIF is "backward compatible". The commonly referred incompatibility issues among version 1.0, 1.1, 1.2, and 2.0 can be summarized to the following issues:

1. The [XML namespace](#) of the Imaging service has changed during the transition from 1.0 to 2.0. This makes Imaging service 1.0 and 2.0 two sets of entirely different commands.
2. The [XML namespace](#) of the PTZ service has changed during the transition from 1.0 to 2.0. This makes PTZ service 1.0 and 2.0 two sets of entirely different commands.
3. There are some structural changes of Device Management service during the transition from 1.0 to 2.0. Among the changes, I/O related commands are moved to Device IO XML namespace in 2.0, while they are in Device Management during 1.0. This changed XML namespace makes 1.0 and 2.0 two sets of entirely different commands.

Other parts of ONVIF specification such as Media service do not suffer from similar issues because XML namespace is not changed. As to the real cause about why make such potentially incompatible XML namespace changes, we think it's because ONVIF is **very prudent** about specification definition. In WSDL, if specification 1.0 defines some XML element to be "mandatory", then it cannot be later changed to "optional" in 2.0, because doing so will violate the principle of backward compatibility. To solve such problem, the most clean way is to create an entirely different XML namespace to separate old commands and new commands, although they look pretty much alike in everyway. If you choose to implement both sets (new and old) of command then you can achieve maximum compatibility. Observing other ONVIF specifications, if it's not absolutely necessary, ONVIF wouldn't choose to use a different XML namespace between old and new versions of specification, and those specifications are readily backward-compatible.

Regardless, if one carefully read and understand those minor differences between versions of specification, it's completely feasible in technical sense to write an ONVIF implementation that supports all versions of specification, or to perform automatic ONVIF version detection.

It's noteworthy that because ONVIF specification is written by human, there can be errors (e.g. bugs). Thus in some minor version specification updates, ONVIF will note the errors from previous version of specification and make relevant corrections, so you can't trust it with 100%.

Just like laws, which requires human to interpret the real meaning behind the text. Interpretation could further change or evolve over time, but in theory it will become more and more clear.

What is XML namespace?

XML namespace is a naming method used by the ONVIF underlying SOAP transport. For example, If we use **timg10** to represent the namespace of Imaging service 1.0, which is "http://www.onvif.org/**ver10**/imaging/wsd", and use **timg20** to represent the namespace of Imaging service 2.0, which is "http://www.onvif.org/**ver20**/imaging/wsd". (Please note the **red part** marks the minor differences between namespaces.) An Imaging service command is called "**GetOptions**" (it's used to acquire the configuration options an imaging sensor supports), then this command in Imaging service 1.0 is written as **timg10:GetOptions**, while in Imaging service 2.0 it's written as **timg20:GetOptions**. Because of the differences in XML namespace, the two functionally identical commands becomes two entirely seperate commands. When developer is writing implementation code, he must write two copies of very similar code to support both versions.

In practice, the minor differences with many similar code could easily cause developers to make a lot of bugs.

As far as I understand, ONVIF only supports video streaming, right? Other functions, such as changing resolution, frame-rate, quality, brightness, motion detection, input signal, output relays, or PTZ aren't defined in ONVIF.

Actually this is **wrong**. For example:

- In [Imaging WSDL specification](#), command **SetImagingSettings** is defined to change Brightness, Contrast, ColorSaturation, and Sharpnes.
- In [Media WSDL specification](#) , command **SetVideoEncoderConfiguration** is defined to change Encoding(Codec), Resolution, Quality, FrameRateLimit, and,BitrateLimit.

Other functions, such as motion detection, input signal, output relays, and PTZ are, in fact, well-defined in the corresponding specifications.

Such misunderstandings can be traced back to the early stage of ONVIF development. At that time, merely supporting video streaming can pass ONVIF test tool and thus declare ONVIF comformance. To simplify matters, most vendors choose to implement only video streaming. All these start to change recently.

Which version of ONVIF specification does Genius Vision support? 1.0, 1.1, 1.2, or 2.0?

We support them all through automatic version detection. Users do not need to know the version details.

I have some questions about "ONVIF verification program" in regards to service terms or business matters, how can I get answers?

This service is just started. Many service terms or business matters need further definition or clarification. Please write us your question in Email and we will respond in **Email** through an single official window.

What is "service terms" or "business matters"? Following are some examples:

1. Can ONVIF verification be conducted indirectly? Can you do ONVIF verification for brand company, instead of an IP camera manufacturer.
2. Can you do ONVIF verification for the same OEM model but with different brands? Can it be done repetitively? Can I apply verification for multiple models at one time?
3. How long does it take to complete ONVIF verification? Does it come with any guarantee or promise?
4. Can I be authorized to use the Genius Vision ONVIF web page materials?
5. Is ONVIF verification service truly free? Will it remain free in the future?

Due to the limitation of human resource, please only inquire important questions, and please allow response delay for processing time.

