HLA Evolved

Frequently Asked Questions

What is HLA Evolved?

HLA Evolved is the working name for the latest update to the HLA Standard. HLA is an official family of IEEE standards given the name IEEE 1516.2000. HLA Evolved will be named IEEE 1516-2010 when it is officially published in 2010.

That means HLA Evolved is really just the latest version of HLA 1516.

How many version of HLA are there?

Like most standards, there are several key versions. Unfortunately, the different versions are not enumerated by a simple number; instead each version has its own nickname. When people talk about 1516, they are typically referring to the 1516-2000 standard or some variation of it. When they talk about HLA Evolved, they are referring to the 1516-2010 standard.

Standard	Nickname	Description
HLA 1.3	"1.3"	A US DoD-managed version of the standard that pre-dates IEEE 1516. Still used widely – particularly in the US. Fully supported by several compliant RTIs (MAK, RTI-NG, Pitch). C++ API supports Dynamic Link Compatibility because major vendors used the same API.
IEEE 1516-2000	"1516"	An IEEE Standard that superseded HLA 1.3 in 2000. The published standard included a C++ API which wasn't useable. There are no implementations of this API. Essentially no RTI's support this.
IEEE 1516-2000 variation	"1516-DoD Interpretations"	US DoD produced a set of interpretations to make the IEEE C++ API work. However, the API does not support Dynamic Link Compatibility - meaning you can't just switch RTIs without relinking or recompiling your federate. One vendor (Pitch) produced an implementation.
IEEE 1516-2000 + SISO-STD-004.1-2004	"1516-DLC-API"	SISO Standardized changes to the IEEE-1516 standard C++ API to fix the original problems and produce an API which supports Dynamic Link Compatibility. Two major RTI vendors support the standard (MAK and RTI NG).
IEEE 1516-2010	"HLA Evolved"	IEEE approved update to the HLA Standard – ratified in 2010. The SISO DLC API was integrated into the IEEE standard along with a number of other changes and improvements. All major vendors have announced commitments to support this API.

Why was there an update?

The standard was updated for two reasons: First, the original standard was broken, and second there were several features the standard didn't support that the community knew it needed.

There were two major problems with the C++ API from the IEEE 1516.2000 standard. First, the C++ API was developed before the C++ language itself was settled. As a result several parts of the original API were not implementable using C++ Standard compliant compilers. Second, the API did not support Dynamic Link Compatibility. Dynamic Link Compatibility makes it really easy to switch among various vendors' RTIs and allows legacy federates to move easily to new federations and exercises.

Additionally, there were a number of features which the community desired which were not present in the standard, including Modular FOMs, Update Rate Reduction, and a better way to manage fault tolerance.

What are the technical changes to HLA?
HLA Evolved made a number of technical changes including:

- **Dynamic Link Compatibility** – That means federates can switch which RTI they use without recompiling/relinking their application.
- **Modular FOMs** – Modular FOMs allow federation developers to break up their object model into useful parts (called FOM Modules). Then each federate only needs to know about the FOM Modules it uses. For example, if a certain 3D Viewer can be controlled by a custom interaction, then the interaction can be turned into a FOM Module. This module can then be used by all federates that wish to control the 3D viewer; Federates not wishing to control the 3D Viewer never need know the module existed in the first place.
- **Update Rate Reduction** – Allows federates to tell the RTI they can only handle data updates below a certain rate. This allows update rate constrained federates to participate in busy federations without bogging down.
- **Better Fault Tolerance** – HLA now has a mechanism for notifying Federates when another Federate becomes disconnected from the network. It means when something goes wrong, everyone will understand what went wrong quickly.
- **WEB Services API (WSDL)** - The HLA Standard now defines a Web Service Description Language (WSDL) "binding". This is similar to the C++, or Java bindings, but for the Web.

For a more detailed look at what's new in HLA Evolved please look here:
http://www.mak.com/pdfs/wp_hla_evolved.pdf

How will MÄK support HLA Evolved?
MÄK will is adding support for HLA Evolved in several phases. First, the MÄK RTI and VR-Link have been released with full HLA Evolved support in 2010. The Logger and VR-Exchange will be released with HLA Evolved support by the end of Q1 2011.

Customers who are under active maintenance for these products will receive the updates as part of their standard maintenance with no additional cost.

Other MÄK products like VR-Forces and VR-Vantage will be released later based on customer needs. If you are interesting in an HLA Evolved version of either of these products please let us know by writing to sales@mak.com.

Does HLA Evolved cost more?

No. When a product is released with support for HLA Evolved, all customers under active maintenance are entitled to it.

Can HLA 1.3 and HLA 1516 Federates communicate with HLA Evolved Federates?

Yes, when you use the MÄK RTI. There is no requirement in the standard for this functionality. However the MÄK RTI uses the same wire format for each version of the HLA Standard. This allows for Federations to exist which contain HLA 1.3, HLA 1516, and HLA Evolved Federates with no requirement for a bridge or gateway.

Do I have to rewrite all my Federates to start using it?

Certainly not! The common wire format (See *Can HLA 1.3 and HLA 1516 Federates communicate with HLA Evolved Federates?*) allows Federation developers to slowly migrate existing Federations to HLA Evolved without incurring large costs. Wire compatibility combined with the ability to use the HLA Evolved features in native HLA 1.3 and HLA 1516 Federates gives federation developers the tools they need to take advantage of all the benefits of HLA Evolved without spending any extra money.

Is HLA Evolved better than other versions?

HLA Evolved is really an incremental upgrade to an already good standard. Although we were able to support the major HLA Evolved features in the MÄK RTI even through the HLA 1.3 and IEEE 1516 APIs, there is something to be said for having these features officially supported in an official IEEE version of the Standard.

Should I switch to HLA Evolved?

That's up to you. MÄK is committed to supporting our customers no matter what version of HLA they use. While you may choose not to immediately upgrade existing Federates to HLA Evolved (See *Do I have to rewrite all my Federates to start using it?*), it probably does make sense to start developing brand new Federates in HLA Evolved.