

# Migrating to HLA 4

With HLA 4 now finalized, and tools like Pitch pRTI 6 supporting the new standard, it's time to discuss how to adopt HLA 4 in real training systems. For new projects, the answer is simple: just start using [HLA 4](#). The architecture and core concepts remain the same as in earlier versions, and new features are easy to incorporate from the beginning.

Most HLA based simulation and training systems are not built from scratch. Instead, they evolve over time and take advantage of one of HLA's strengths: the ability to reuse components, systems and simulators across generations of development. The interesting question is, how can HLA 4 be integrated into these existing environments?

This article explores practical strategies for adopting HLA 4, focusing on incremental improvements and long-term system evolution.

## Local changes and Federation changes

When planning an upgrade to HLA 4, it's useful to separate improvements into two categories:

- Local changes are modifications to a simulator or federate that do not require any updates to the federation agreement. These changes are internal to one system and do not affect how simulators interact. They are safe to implement independently.
- Federation changes affect how simulators interact and exchange information. These updates require modifications to the federation agreement and often need to be coordinated across multiple systems. They should be planned and introduced carefully.

Understanding this distinction helps prioritize changes that can be adopted quickly while planning more complex updates over time.

## Starting with the Object Model

One of the most effective ways to begin adopting HLA 4 is through updates to the Federation Object Model (FOM). Tools like [Pitch Visual OMT 3](#) support automatic loading and conversion of older FOMs to the HLA 4 format, making it easy to take advantage of the new object model features right away.

Several object model enhancements are local changes:

- Reference data types: You can now define relationships between objects and interactions explicitly. This was previously only expressed in naming and semantics. Now it's part of the model and machine-readable.
- Required attributes: You can mark attributes as required for specific object classes. This makes it clear which values simulators should provide when creating instances.
- Switch default values: Previously, switch settings were required but not well understood. HLA 4 introduces sensible defaults, allowing most users to ignore them unless a specific setting is needed.
- DDM improvements: Data Distribution Management is now fully specified, including dimension data types. This removes ambiguity and makes it easier to implement effective sender-side filtering.
- Extended merge: One of the headline features in HLA 4. It lets you add attributes, parameters, and DDM dimensions to existing classes. This was previewed in Pitch pRTI back in 2019 and is now production-ready.

These changes do not require modifications to existing federates or federation agreements. They can be adopted incrementally and independently.

## Federation level Enhancements

Other HLA 4 features represent federation changes, meaning they affect how simulators coordinate and interact.

- Directed interactions: Interactions can now be sent to specific object instances, which reduces unnecessary network traffic and simplifies tasking. For example, you can now issue an order directly to a simulated unit rather than broadcasting to all listeners.
- Unsigned data types: A long-requested feature, now officially supported. This simplifies integration with other models and reduces error potential.

Implementing these features requires updating the federation agreement and modifying any simulators that will use them. They bring powerful capabilities but must be introduced with coordination.

## The updated API and Developer Experience

HLA 4 has an updated and modernized API. [Pitch pRTI 6](#) supports this interface, and it includes updates that simplify development while improving consistency.

Here are the key points:

- C++11 and Java 11 or later are required.
- Namespace and package names have changed from “2010” to “2025.”
- Method signatures have been updated. Most of these are minor, and the compiler will help you spot them.
- To handle callback changes, use override annotations in your federate ambassador so the compiler can verify correctness.

If you’re using Pitch Developer Studio, things get even simpler. Differences in HLA API versions are handled under the hood, and the upcoming version 7 will support HLA 4. This enables developers to upgrade without needing to manually refactor their codebases.

If you plan to take advantage of HLA 4-specific features in your federation agreement, you’ll need to make corresponding updates in your simulators:

- Directed interactions require new service calls in the API.
- The Encoding Helpers have been expanded to support:
  - Unsigned data types.
  - Extendable enums.
  - Extendable variant records.
- Time management updates:
  - Federations can now require at least one regulating federate in a time managed federation.
  - A new service to provide more control of time advancement.

## Additional Quality-of-Life Improvements

HLA 4 isn't just about interoperability and performance. It also includes improvements that make life easier for developers, integrators, and operators. These are all local changes and can be adopted incrementally:

- The producing federate is now always available.
- You can query ownership for multiple attributes at once.
- A new RTI Configuration object replaces the old connection string for a more flexible setup.
- New services include reporting when a federate has resigned and listing all federations.

These features help streamline integration, debugging, and federation management.

## Security and Deployment

[Security](#) is a growing priority in distributed simulation. HLA 4 introduces new deployment options that support encryption and authentication:

- The [Federate Protocol](#) supports TLS, including mutual TLS, and secure WebSockets.
- Federate authentication restricts access to specific RTIs or federations.

These features are part of HLA 4's formal specification and are supported in tools like Pitch pRTI. They enable secure deployment without requiring external frameworks. If you're working in an environment with strict security requirements, this is a game-changer.

## Moving Forward

HLA 4 brings significant benefits while preserving what already works. Whether you're starting fresh or modernizing existing systems, there's a clear path forward.

Start by identifying which safe, local changes you can apply right away. Then plan for larger updates involving federation agreements and simulators. With the right approach, you can adopt HLA 4 without disruption and start reaping the benefits immediately.

If you're preparing your system architecture or planning upgrades, now is the time to begin using HLA 4. Pitch Technologies has the expertise, tools, and proven migration strategies to help you transition smoothly, whether you're starting fresh or upgrading a long-standing federation.