



- Are you looking for a phased component upgrade, all-inclusive upgrade, or a quick upgrade solution?
- How important is it to maintain your integration mappings and code base during the upgrade?
- Do you want to preserve your canonical documents, shared objects, or internal naming conventions?
- Are you confused about how to upgrade Trading Networks, EDI documents, custom adapters, custom code, adapter-specific operations, or Modeler?
- How important are web services, SOAP, and technology standards to your integration solution?

webMethods 6 Upgrade

There are multiple Upgrade approaches: What's the best methodology for my environment?

There are multiple upgrade implementation methodologies; choosing the most suitable one depends on your current integration environment, future integration plans, product availability, and resource constraints.

In order to help our clients choose the appropriate upgrade methodology, Visual Integrator Consulting first performs an Upgrade Assessment. The Upgrade Assessment reviews the environment, each integration object, and usage patterns to determine the integration's complexity, re-usability, naming conventions, and product dependencies. The Upgrade Assessment deliverables include a detailed complexity rating for each integration object, recommended methodology & justification for each integration, and high-level project plan for the upgrade.

Visual Integrator Consulting's Upgrade Best Practices, which are in use at many webMethods 6 customers, complement our defined upgrade implementation methodologies.

What are my Upgrade options?

Re-code the integrations in webMethods 6. While this can seemingly be the most complicated and time intensive method, it often makes sense since it affords the greatest flexibility. There are no limitations on naming conventions, use of 6.x features, and will be 100% Web Service compliant.

webMethods 6 Upgrade Utility. webMethods' upgrade tool is designed as an aid to the planning and implementation of a 6.x upgrade. This tool is most effective in analyzing code from off-the-shelf adapters. Typical upgrades achieve 30-70% timesavings by utilizing the upgrade tool. There is little flexibility in naming conventions, hierarchical structures, and code interpretation from the upgrade outputs.

Phased approach. This methodology involves upgrading various architectural components to v6, while others remain in a non-v6 environment. This is a common approach regardless of the individual upgrade strategy used. A phased approach is important to customers that have frequent modifications to existing code. This also allows new initiatives to quickly take advantage of the new 6.x environment.

Dual Mode Operation. Some integration environments are not well prepared for a webMethods 6 upgrade or would not substantially benefit from an upgrade. The webMethods 6 product is run-time compatible with prior versions. Occasionally, it is best to leave the existing environment unmodified and use 6.x for new work.

Upgrade Considerations

We recognize that no two implementations are the same. Likewise, no upgrades are exactly the same. Visual Integrator Consulting consultants have experience upgrading a variety of areas related to your integration implementation. Below is a list of some factors, which influence the proper upgrade methodology.

- Adapter 6.x availability/compliance
- Complexity of the integration code
- Naming Conventions
- Web Services & Standards
- webM 6 features importance within architecture
- Performance & latency requirements
- Allowable Production downtime
- System Stability
- Personnel retraining on IDE
- Amount of custom code
- Time and Effort

How do I get started?

Call Visual Integrator Consulting or visit our web site for more information. We'll give you a fixed-bid for your webMethods 6 Upgrade Assessment so you'll know the cost before work begins. You are under no further obligation to Visual Integrator when it's complete.

Let us take the pain, uncertainty, and anxiety out of your Upgrade process. Call us today.