When you use an automated tool to facilitate tracking dependencies between requirements and artifacts (in this case, Rational Application Developer for WebSphere Software), you can obtain a project-wide omniscient view of all dependency links, and that’s a good thing.

In a development project, dependencies between different requirements, as well as requirements and software artifacts, can be difficult to track and assess their impact due to the large number of them. The "perfect storm" of dependency challenges includes:

- **Applications with large numbers of dependencies** Many libraries that require long downloads that might not be very portable.
- **Lengthy chains of dependencies** $\text{liba}$ depends on $\text{libb}$ and so on.
- **Dependencies that conflict** This can affect on-the-fly installs because installers might not load different library versions at the same instance. Sometimes you can see a serious version of this problem in Linux systems: Installing packages from different distributors can create a chain of dependencies that can conflict with the critical $\text{glibc}$, forcing you to then uninstall thousands of packages.
- **Dependencies that rely on a circular chain of resources** $A$ needs $B$ which needs $C$ which needs $A$; in some platforms you can resolve this by simultaneous install of all resources, but on some platforms you cannot resolve it.

"Rational Application Developer's seamless integration with Apache Cordova lets you trace complex mobile code dependencies, ensuring rapid conversion of existing web apps into hybrid mobile ones."

Being able to track and assess the impact of requirement and artifact dependencies is a critical skill a developer needs in order to take efficient control of a development project. Automated tools can help the developer assume that skill quickly and easily.

Part of the task of organizing dependencies in a successful development project is to make and manage the following types of associations between requirements and domain elements:
You can trace dependencies between requirements and artifacts easily. **IBM® Rational® Application Developer for WebSphere Software** makes following the association links between a requirement and other artifacts, even complex mobile artifacts, easier. Give it a try. **Register for the 60-day free trial of Rational Application Developer** and build some reusable association rules or model a use case from an old project, create a sample project and explore dependencies, or start a new mobile app conversion project from the requirements up.

- A **direct association** is a link that associates a domain element and a requirement and synchronizes both artifacts so that you can manage them as a single, conceptual item. This is fairly simple to make and manage manually, unless your project contains a lot of these one-to-one associations.
- An **indirect association** links domain elements and requirements through proxy requirements and traceability. Indirect associations are particularly useful for creating many-to-many relationships with traceability between multiple requirements and a single proxy. They are also useful between multiple proxies and a single requirement. Thanks to the concept of code reuse, most projects generate these sorts of relationships; the many-to-many nature of these relationships is most easily managed using an automated platform.

An automated tool takes a lot of the work out of performing the following development tasks:

- Establishing traceability between requirements and artifacts to indicate dependencies or to establish that you've satisfied requirements.
- Monitoring the impact of change within your project, as well as synchronizing changes project-wide.
- Developing rules based on associations to further automate requirement changes; also developing and modeling use cases for later reuse by viewing entire association trace routes.
- Accumulating multiple associations in packages or folders in order to relate entire groups of artifacts with a specific requirement.

For more information on tracing dependencies (especially between development project requirements and corresponding coding components), see the article "[Tracing project requirements](#)"; it will introduce you to another trial platform you can experiment with in order to automate dependency tracking and assessment.

**Try Rational Application Developer for free**
**Learn more about dependency tracking**

In a software development project, keeping track of complex dependencies is a critical and involved, yet not-so-creative task. It is also a task that lends itself to low-level automation in order to discover and understand the impact of changes to code with dependencies. An automated tool can remove much of this burden and allow you to concentrate on building an imminently elegant, functional application in the shortest time possible.
Related topic

- IBM Rational Application Developer for WebSphere