

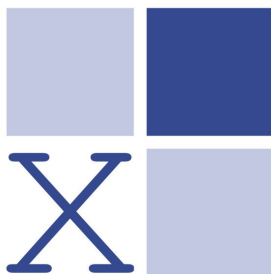


Manage Use of External Libraries

**Using Lattix™ LDM™ to enforce your 3rd Party
library adoption process**

**Whitepaper
January 2005**

Copyright © 2005-7 Lattix, Inc. All rights reserved



Lattix for Software Architectures

The Lattix LDM™ solution employs a unique and powerful strategy for communicating, visualizing, analyzing, and managing a software system's architecture. The Lattix approach allows the architect to formalize and enforce the overall architecture of a system through design rules, and to leverage DSM as a key component of Lattix's approach for controlling the complexity of a large software system. Further, this architecture is easy for managers to understand which makes it easier for business issues to be incorporated in the development process.

Using 3rd Party Libraries

Whether including open source components or purchased 3rd party components, most large software systems being built today leverage some form of 3rd party components. As a result, most organizations need some level of process to ensure that the licensing requirements of the 3rd party package are understood and met. For instance, when and where to credit or include copyright information for the 3rd party component is frequently specified in the license agreement.

However, these processes are typically manual, relying on development staff to follow in a timely manner, rather than an automated mechanism to check for compliance. Often, this means that newly included 3rd party libraries are assessed only at the end of the software release cycle, or worse, at some point after the software release.

With Lattix solutions, you can establish an automatic "gate keeper" for 3rd party libraries. That is, you can create Design Rules that allow use of approved 3rd party libraries, and by default disallow the use of non-approved libraries. Moreover, you can control which of your systems/products can use which 3rd party libraries

Automatically Extracted Usage

As part of building a dependency model for your system, LDM extracts the usage information from the input. For Java it extracts all usage information from Java classes – this includes all references to classes which are in third party libraries and may not even have been loaded into the project.

Because these dependencies can be extracted quickly, LDM makes it easy to check for new dependencies on un-approved 3rd Party libraries and flag them as violations.

Creating and using Design Rules for external libs

LDM allows you to explicitly allow or disallow dependencies between subsystems including between your subsystems and external subsystems. These are called Design Rules. These rules allow the architect to formalize their architectural intent, and to codify that architectural intent with rules such as ones that permit references or disallow references between subsystems. With LDM, the architect can establish, and enforce, rules that control external library use as well as those that reflect intelligent abstraction, appropriate layering, and componentization. For example, if an architect wanted to enforce that all database access pass through a single module, they could easily do so with Lattix design rules.

External Library Design Rules

The system designer or 3rd party library administrator can establish rules for external library use at the beginning of a project before code is written or can create rules for an already existing software system.

By default, the system can use itself

By default, the system can use java and javax

External libraries are disallowed until explicitly allowed through a design rule

Rule	Target
Can-Use	⊕ \$root
Can-Use	⊕ java
Can-Use	⊕ javax
	⊖ org
	⊖ org.apache
Can-Use	⊕ org.apache.log4j
Can-Use	⊕ org.apache.oro
Can-Use	⊕ org.apache.regexp
	⊖ org.w3c
Cannot-Use	⊕ org.w3c.dom
	⊖ org.xml
Can-Use	⊕ org.xml.sax
Classifications	

Can Use
Cannot Use
Delete Rule
Move Up

Move Down
Add Rule
Edit Rule
Save Rule

Design Rule allows the system to use org.apache.log4j

Design Rule disallows the system to use org.w3c.dom

Figure 1. LDM Design Rules editing for external libraries

In the depicted example, the system, *org.apache.tools.ant*, makes use of a number of packages from *org.apache* and *org.w3c*. In the depicted rule editing session, the system *org.apache.tools.ant* is allowed through “Can-Use” rules the following subsystems: *org.apache.log4j*, *org.apache.oro*, and *org.apache.regexp*. However, the system is not allowed to use *org.w3c.dom* through a “Cannot-Use” rule.

For more details on LDM Design Rules, and how they can manage your system’s architecture as well its use of 3rd party libraries, please see the Lattix whitepaper, *Design Rules to Manage Software Architecture*.

Checking for violations

Lattix solutions include the ability to check for rule violations from LDM's user interface or in a nightly build or as part of a code submission process using LDC. Violations can be interrogated from the LDM's User Interface, or a log can be produced from LDC.

Some example output showing violations for parts of *org.apache.tools.ant* that use *org.w3c.dom* are shown in Figure 2.

```
Violations for org.apache.tools.ant.XmlLogger:
  org.w3c.dom.CDATASection
  org.w3c.dom.Element
  org.w3c.dom.Node
  org.w3c.dom.Document
Violations for org.apache.tools.ant.taskdefs.XmlProperty:
  org.w3c.dom.Node
  org.w3c.dom.NodeList
  org.w3c.dom.Document
  org.w3c.dom.Element
  org.w3c.dom.NamedNodeMap
```

Figure 2. LDC report output of External library use violations

Besides design rule violations, the LDC command line program also reports overall statistics for number of classes and dependencies for the project, and records classes that have been removed from the system, and classes that have been added to the system.

Note: This can be used to monitor new classes coming into the system to see if they match development activities.

Incorporating Lattix into your 3rd party adoption process

Lattix solutions make it easy to codify your rules regarding 3rd party library use, and then to set up a monitoring system that allows you identify a new 3rd party library when it is has been added to your software system instead of late in the release cycle or after the release cycle. The steps to augment your 3rd party adoption process such that Lattix solutions can act as an automatic "gate keeper" for 3rd party libraries are:

1. Load your software system into LDM
2. Create Design Rules for which 3rd party libraries have been approved by your development organization
3. Setup the build process to email LDC logs from nightly build to your appointed 3rd party library administrator.