



Re-Approaching the Refactoring Rubicon

1. Refactoring's Rubicon: Extract Method

- Automating **Extract Method** is Refactoring's Rubicon (Fowler*)
 - The one that demonstrates "serious tool support"
 - Precondition for many other transformations
 - **This Rubicon has not yet been crossed**
 - Getting it right requires more analysis capability than is available in current IDEs
- * <http://www.martinfowler.com/articles/refactoringRubicon.html>

3. Currently Unsupported Cases of Extract Method

(a) Extract multiple fragments

```
response.setContentType("text/html");
disableCache(response);
String albumName =
    request.getParameter("album");
PrintWriter out = response.getWriter();
```

(b) Extract a partial fragment (select sub-expressions as parameters)

```
out.println(DOCTYPE_HTML);
out.println("<html>");
out.println("<head>");
out.println("<title>Error</title>");
out.println("</head>");
out.print("<body><p class='error'>");
out.print("Could not load album '" +
    albumName + "'");
out.println("</p></body>");
out.println("</html>");
```

(c) Extract loop with partial body (loop duplication with data flow)

```
out.println("<table border=0>");
int start = page * 20;
int end = start + 20;
end = Math.min(end,
album.getPictures().size());
for (int i=start; i<end; ++i) {
    Picture picture = album.getPicture(i);
    printPicture(out, picture);
}
out.println("</table>");
```

(d) Extract code with conditional exits

```
if (album == null) {
    new ErrorPage("Could not load album '"
        + album.getName() + "'").printMessage(out);
    return;
}
```

2. A Case Study in Enterprise Refactoring

- Converted a Java Servlet to use the MVC pattern*
- Used as much automated support as available
 - The whole conversion could be described as a series of cataloged ("small") refactorings
 - Most steps were inadequately supported by the IDE
 - Some were not supported at all (see the tables below)

* Based on Alex Chaffee's "Refactoring to Model-View-Controller" article (<http://www.purpletech.com/articles/mvc/refactoring-to-mvc.html>)

Fully Supported Refactorings	Uses
Extract Method	3
Extract Temp	3
(Self) Encapsulate Field	2
Replace Magic Number with Symbolic Constant	1
Inline Temp	1
Extract Superclass	1
Delete Methods	1
Move Method	1
Total	13

Partial(*) or No(**) Support	Uses
Extract Method *	10
Substitute Expression **	5
Replace Temp with Query *	3
Replace Method with Method Object **	2
Substitute Statement **	1
Extract Class *	1
Move Statement (or Swap Statements) **	1
Total	23

4. Crossing the Rubicon: Requirements for Big Impact

