

# Grammatical extensions to Java and AspectJ for Explicit Join Points

TECH REPORT ejp-200703-02

Kevin Hoffman  
Purdue University  
305 N. University Street  
West Lafayette, IN 47907  
kjhoffma@cs.purdue.edu

Patrick Eugster  
Purdue University  
305 N. University Street  
West Lafayette, IN 47907  
peugster@cs.purdue.edu

## 1. SYNTAX EXTENSIONS

Herein we formally define our extensions to the AspectJ and Java grammars. Our grammar extensions are LALR(1), and their descriptions follow the conventions used in the Java language specification.

### Syntax added to AspectJ:

---

*ExplicitJoinPointDeclaration:*  
*ExplicitJoinPointHeader FormalPointCutParams<sub>opt</sub> Handles<sub>opt</sub> Throws<sub>opt</sub> ExplicitJoinPointInitializer<sub>opt</sub>*

*ExplicitJoinPointHeader:*  
*Modifiers<sub>opt</sub> scoped<sub>opt</sub> ExplicitJoinPointDeclarator*

*ExplicitJoinPointDeclarator:*  
*joinpoint ResultType Identifier (FormalParamList)*

*FormalPointCutParameters:*  
*pointcutargs FormalPointCutParameterList*

*FormalPointCutParameterList:*  
*FormalPointCutParameter*  
*FormalPointCutParameter, FormalPointCutParameterList*

*FormalPointCutParameter:*  
*Identifier (FormalParameterList<sub>opt</sub>)*

*Handles:*  
*handles ClassTypeList*

*ExplicitJoinPointInitializer:*  
*= Expression*

-----

*MethodConstructorPattern:*  
...  
*thisclass || thismethod || thisblock*  
*ExplicitJoinPointPattern*  
*ExplicitJoinPointScopePattern*

*ExplicitJoinPointPattern:*  
*ejp (MethodPattern)*  
*!ejp (MethodPattern)*

*ExplicitJoinPointScopePattern:*

*ejpScope (MethodPattern)*  
*!ejpScope (MethodPattern)*

-----  
*DeclareDeclaration:*

...  
*declare error nomatch Granularity : PointcutExpression :*  
*PointcutExpression : StringConstant*

*Granularity:*

*by package*  
*by class*  
*by method*  
*by ejp*

---

### Syntax added to Java:

---

*ExplicitJoinPointExpression:*  
*ExplicitJoinPoint PointcutArguments<sub>opt</sub> Block<sub>opt</sub>;*

*ExplicitJoinPoint:*  
*AspectName . ExplicitJoinPointName (ArgumentList<sub>opt</sub>)*

*PointcutArguments:*  
*pointcutargs PointcutArgumentList*

*PointcutArgumentList:*  
*PointcutArgument*  
*PointcutArgument, PointcutArgumentList*

*PointcutArgument:*  
*PointcutName (FormalParamList<sub>opt</sub>) : PointcutExpression*

---