



Fables FAQ

Prepared by:
Viktor Erdélyi, Iványi Dávid Márton and
Richárd Olivér Legéndi



November 2007, Budapest

1 Contents

1	Contents	2
2	General	3
3	Syntax problems	5
4	Scheduling	8
5	Randomization	9
6	Charting Wizard	10
7	Linux problems	12
8	Keeping the IME up to date	13
8.1	Upgrading the IME	13
8.2	Integrating the Fables plug-in to an existing Eclipse installation	16
8.3	Disabling, uninstalling and switching between plug-in versions	17
9	Appendix	18
9.1	The Fables EBNF grammar	18

2 General

1. ***Q: What do I need to run the Fables Integrated Environment (IME)?***
A: You'll need the Java(TM) SE Runtime Environment 6 or later. Repast and additional libraries are all included in the installation package.
2. ***Q: What is the latest version of the Fables IME?***
A: The latest version is v1.45, you can find updates at <http://kysss.inf.elte.hu/fables>.
3. ***Q: Is the Fables IME written in Eclipse?***
A: No, the Fables IME is an Eclipse product with several Eclipse Plug-ins included, like the Fables plug-in and other plug-ins needed to compile, run and handle generated Java sources.
4. ***Q: Can I download Fables as a plug-in for my Eclipse installation?***
A: Yes, it is possible to download the Fables plug-in, and integrate it into an existing Eclipse installation. See details in 8.2.
5. ***Q: How do I report a bug/request a feature in the Fables IME?***
A: To manage and track IME bugs and feature requests, the project uses Bugzilla. The main entry point to the Bugzilla can be found at <http://kysss.inf.elte.hu/bugzilla>. Please post a bug if you see something wrong, even if you are in doubt whether it really is a bug.
6. ***Q: Where can I find more information about the Fables compiler/IME?***
A: The official Fables site is <http://fables.aitia.ai>. Another site is used for announcements and upgrades, which is <http://kysss.inf.elte.hu/fables>. If you have further questions, either you can contact us via e-mail at legendi@aitia.ai or visit our IRC channel at <irc://irc.freenode.net/#fables>.
7. ***Q: How can I learn Fables?***
A: We advise to:
 - Start with the Fables tutorial included with the software to understand the basics of both the language and the IME.
 - Take a look at the Fables user guide to get familiar with the language syntax and the use of the IME.
 - The Fables IME has many built-in example applications, examine them, and explore the possibilities of the language.
8. ***Q: The Fables compiler is written in Java, isn't that too slow?***
A: No, Java is often used for writing compilers. The best example is Sun's javac compiler, which is written in Java. Sun's java JVM interpreter is mainly written in Java too (except some platform-specific and low level routines). Sun's Java Web Server is also written in Java.
9. ***Q: Which Repast version is currently supported by the Fables IME?***
A: The Fables compiler generates source for the latest stable Repast J version (3.1).
10. ***Q: When I click either on the Fables icons at the toolbar (e.g. the Run Fables code button), it says "Please select and open a valid fabfile!". Which model is going to be processed?***
A: The quick launch icons process the Fables model in the currently active editor. You only have to open the model you wish to process.

11. ***Q: Can I contribute to the Fables demos? What do I have to do?***

A: Yes, we would be happy to include any Fables models you've written in the sample models! You only have to send us the Fables sources, the created chart xml file and a short description.

12. ***Q: How is the Fables IME licensed?***

A: Fables IME is free to use for everyone but is not an opensource application. See the license on our webpage or *license.txt* in the package.

3 Syntax problems

13. **Q: What is the syntax definition of the Fables programming language?**

A: You can find the EBNF grammar description of the language attached in appendix 9.1.

14. **Q: I'm getting "Unexpected token: ..." errors. What does that mean?**

A: The most possible reason is that the compiler fails to parse an expression due to syntactical problems. Check if there's a ';' character at the end of each line, whether the bracketing is correct and the time events are defined for the schedules. These are the most common reasons.

15. **Q: I'm getting "Unresolved function call: ..." errors. What does that mean?**

A: It means that the compiler cannot recognize a function call in one of the expressions. The most common reason for that is that it can't determine the parameter lists in a compound function call, like: `function1 function2 parameter`. The compiler greedily parses the parameters, and the outer function will have all the parameters if it isn't grouped by brackets. The solution is to use correct bracketing. Here are two examples for the explanation: let's assume that there's a `function` which returns a set or sequence result. The function `fsum` is to sum these values. The first definition is wrong, because the compiler thinks that there's a function (+) which accepts a function, and a tuple of (x, y) as parameters.

Wrong definition: `fsum = (+) function (x, y) ;`

Correct definition: `fsum = (+) (function (x, y));`

To avoid these kind of errors, please always use correct bracketing for function expressions.

16. **Q: What's the difference between the many = operators?**

A: We have 3 different operators similar to =, the `=`, `==` and the `:=` operators. These are similar in their forms, but they do have their own meaning and places where are usable.

- *The definition symbol (=) can be used only for defining own functions or constants;*

```
worldSize = 50;
```

- *The assignment operator (:=) can be used for assigning a value to a variable you have declared. It is mostly used when you create a new instance of an agent to set up its initial parameters/behavior;*

```
new Agent[ age := 0, id := 10 ];
```

- *The equals operator (=) is a logical operator used to determine whether two items are equal or not (this is the opposite of the <> operator);*

```
println( 1 == 2 );
```

```
→ false
```

17. **Q: How do I make a model parameter I can change during runtime?**

A: Create a constant, and declare a default value to it. Then just put it in the model's `startUp()` section to the parameter list like below:

```
model Params {
```

```

myParam = 5; // Parameter's default value

// Parameter list
startUp( myParam ) {
    println( "My parameter is currently: " ++ myParam );
}
}

```

18. **Q: What model members can be used as parameters? Can classes also have parameters?**

A: In the current version it is only permitted to use logical and numeric parameters. Classes' `startup()` sections cannot have parameter lists, it leads to compilation error.

19. **Q: I've declared a numeric parameter, but when I set its value to real, it seems it doesn't change. What's the reason?**

A: The most common reason is the parameter is evaluated as integer by the Type System. Thus, it cannot have a real value (the compiler rounds every assignment downwards, i.e. 0.9 is rounded down to 0). Let's say you have the following model:

```

model Params {
    param = 0 ;

    startUp( param ) {
        println( param ) ;
    }
}

```

When you run the model above, and try to set the value of `param` to 0.9, you'll see its value remains 0. To avoid this sort of errors, declare `param` as a real constant like this:

```

param = 0.0 ;

```

20. **Q: Where do I use the `where` and the `when` keywords? Aren't they the same?**

A: No, they have different meanings.

- The `where` keyword is used for defining a complete expression (a model or class member with local constant definitions). For instance, here's a valid usage of `where`:

```

x = [ 1, 2, 3, 4, 5 ] ;
meanAverage = sumX / size(x) where ( sumX = sum(x) ) ;

```

- The `when` keyword is used for defining a logical condition for a set or sequence type after the iterations:

```

rndset = { i : i is [1..10] when uniform(0,1) < 0.5 } ;

```

The example set above contains each element of the interval [1,10] with 50% probability.

21. **Q: I want to print a sequence or set, but instead of the elements I see only something like [I@9ab0. What's that?**

A: The `print` and `println` functions can only handle two dimensional sequences. When using these functions on higher order sequences, it will simply call the default Java `print` command, which results in this weird String

(the `[]` prefix means that this is an array, and the substring after the `@` character is the most unique hash value of the object).

22. **Q: How to acquire the (i,j)th element of my 2 dimensional matrix?**

A: The correct form of indexing is `mx(i)(j)`, not `mx(i,j)` – where `mx` is a two dimensional matrix. The get operator is a function too, that has exactly one parameter.

23. **Q: I've created a lot of Agents, but I can see only one of them! Why?**

A: When you create a new agent instance, it is stored in the set of all agents. This set can be referred to with the name of the agent class. If you create more than one instances of the same agent, it will be constructed, but in the simulation, there'll only be one. Take the following example:

```
class Agent { id = 2; }

startUp {
  [ new Agent : _ is [1..5] ];
  println(Agent);
}
```

The output is:

```
{ Agent {id=2} }
```

Important: Agents are equal to each other if all of their variables and constants have the same values.

Using a unique ID attribute will solve the problem:

```
class Agent { var id; }

startUp {
  [ new Agent[id:=i] : i is [1..5] ];
  println(Agent);
}
```

The output will be:

```
{ Agent{id=1}, Agent{id=2}, Agent{id=3}, Agent{id=4},
  Agent{id=5} }
```

4 Scheduling

24. *Q: I've removed every schedule from my simulation with `removeSchedule()` function, but the simulation doesn't stop. Why?*

A: In the generated *Repast* code the declared schedules are only `BasicActions`. We use one global scheduler to merge and process each action.

The simulation doesn't stop running because it can't be determined whether a `Schedule`'s `ActionQueue` is empty or not.

Use the `pause()` and `stop()` functions for interrupting the simulation.

25. *Q: I've examined the generated Java model, but I see only one Repast Schedule object, although I've defined several ones! Is that correct?*

A: Yes, that's absolutely correct. The compiler hides the real schedules you've defined, and merges them into one schedule. Both named, unnamed, cyclic and acyclic schedules are handled differently, but in general their first activation turn, cycle and actions are all stored in this general schedule.

26. *Q: Where's the `startUp()` section I defined in the generated code? I don't see any functions like that.*

A: The `startUp()` section of a Fables model is called `begin()` in the generated *Repast* code. All statements you've written in the `startUp()` section have their generated code there. Additionally, you can spot some other calls you haven't written in the Fables source, like variable and schedule initializations. These are all necessary to make the model work properly.

27. *Q: Is there any difference between the named and unnamed schedules?*

A: Basically there's no difference between them. The benefit of using named schedules is that they can be re-scheduled or stopped dynamically during the simulation.

28. *Q: There's a function called `addEvent()`, does it have any relation to schedules? Can I add an event to a running schedule with it?*

A: No, not yet. At the moment, `addEvent()` can only process a statement (any function call without a real return value, like the `println` function call or an assignment) at the given simulation turn.

5 Randomization

29. *Q: Function `discreteUniform(a,b)` doesn't return any values between *a* and *b*!*

A: Yes, that is the standard behavior of `discreteUniform()`. For instance, `discreteUniform(1,3,4,5,8,10)` will return one of the values 1, 3, 4, 5, 8 or 10. To make it choose a number randomly from an interval from *a* to *b* use the expression `discreteUniform([a..b])`.

30. *Q: How do I use the same random number repeatedly in the same definition?*

A: You can use the same random value in a function definition with local definitions like the following:

```
rnd(i) = (act <> i) => act otherwise rnd(i)
      where ( act = discreteUniform([a..b]) );
```

Please note that local definitions cannot have parameters at the moment, so a definition like

```
wrnd_def = localFunc(1)
      where (
        localFunc(x) = 2*x; // This is illegal definition
      );
```

is illegal and causes compilation errors.

6 Charting Wizard

31. **Q:** *I can zoom in to several charts (i.e. the TimeSeries chart) if I select a rectangle area on the chart with the mouse while pressing the left mouse button. How do I zoom back?*

A: Hold down the right mouse button while the mouse cursor is above the zoomed chart, move the mouse to the left and upwards and release the right mouse button. This will restore the default view of the chart.

31. **Q:** *Is it possible to have more than two grid charts on a composite grid?*

A: In the future versions we're planning on having a universal grid chart that can have any number of grid charts using any kind of data sources. At the moment the composite grid can only be used to compose two grid charts, and only with the same kind of data sources.

32. **Q:** *I want to create a composite grid chart with random filling. I have different kind of data sources: one of them is a matrix that I want to use for the color values in the background, while I get the x and y coordinates from sequences. I can see both of them in the first list, but when I select one of them, the other one disappears from the second list! Can I create a chart from these different data sources?*

A: No, unfortunately the composite grid chart accepts only the same kind of datasources at the moment. At the moment little workaround has to be done to achieve this task. You have to explicitly convert one of the datasources to the other one with a function. See the following example:

Let's say you want to create the background using a color matrix and get the shape coordinates from two sequences (the sequence of X and Y coordinates). The definitions would be something like this:

```
colors_matrix =
  [ [ discreteUniform (0, 1)
    : i is [ 0 .. 2 ] ]
    : j is [ 0 .. 2 ] ] ;

x_coords = [ 0, 1, 2 ] ;
y_coords = [ 0, 1, 2 ] ;
```

You have to use the following definition to convert the `x_coords` and `y_coords` to a matrix:

```
x_y_matrix =
  [ [ ( i in x_coords and j in y_coords ) => 1
    otherwise 0
    : i is [ 0 .. 2 ] ]
    : j is [ 0 .. 2 ] ] ;
```

33. **Q:** *I've seen there are several statistical custom producers. Is it possible to use them over collections?*

A: No, currently the statistical producers can only be used for one-dimensional collections (sequences or sets).

34. **Q:** *I've created a Network chart and the Nodes of the network are always moving between the simulation turns, even if I set the network's layout to Circle Layout. Am I doing something wrong? Is it possible to make the nodes stay in place?*

A: **No, that's a known issue.** We are working on the solution. Unfortunately there's no way to make the network's nodes stay in place yet.

35. *Q: I want to create a user-defined producer for the width and height values, or color values, but I cannot see the "Advanced" settings button where I could set the properties. Where do I do that?*

A: **That's a known issue.** Currently there's no option for using a user-defined producer as width and height properties.

36. *Q: I have several charts and I want to use more than one user-defined producers. After editing one's properties, it has disappeared from the list! Can't I have more than one user-defined producer?*

A: The short answer is that you can have any number of user-defined producers, but you can only create one when editing a chart. The reason is that creating some of the custom producers can be a source for a lot of errors, since the syntax and meaning of the given script can't be checked (there's no source generated yet). The generated source should be and is reviewed each time to make sure there's no problem in the generated custom producers. To create more user-defined producers click on the "Edit chart" button in the toolbar: each time you'll be given a new user-defined producer.

7 Linux problems

37. *Q: When I try to start the IME on Linux (especially on Gentoo-based distributions), I get errors like "xcb_xlib.c:50: xcb_xlib_unlock: Assertion 'c->xlib.lock' failed." What can I do about it?*

A: This is a known compatibility problem between xcb and Java. You have two options to resolve this:

- Run the following command as root (Gentoo-based distributions only):

```
CFLAGS="${CFLAGS} -DNDEBUG" emerge -alv xcb libxcb
```

- Locate your Java installation path, and execute the following command as root, replacing the path with yours:

```
sed -i 's/XINERAMA/FAKEEXTN/g' /usr/lib/jvm/java-6-sun-1.6.0.00/jre/lib/i386/xawt/libmawt.so
```

Note: for more information, please visit the following page: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6532373

38. *Q: I get a dialog box with a huge error message when I try to start the IME on Gentoo-based Linux distributions. What's the problem?*

A: You need to start the IME with the following command (or you can create a script that does this for you):

```
#!/bin/bash
./fables -vm `java-config --java`
```

Replace "./fables" with the proper path.

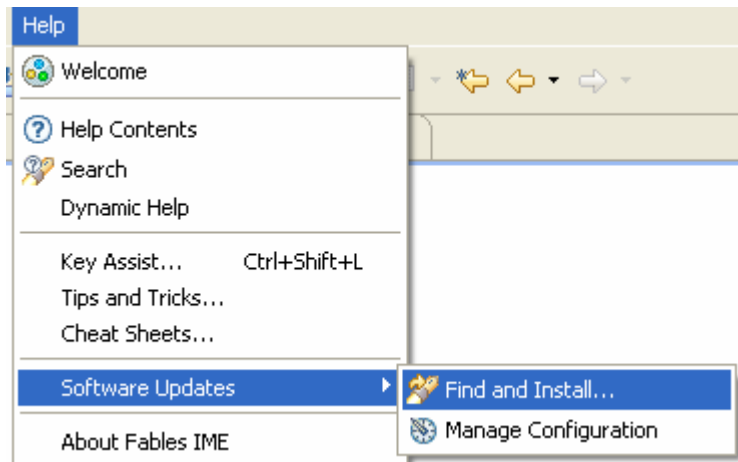
8 Keeping the IME up to date

Both the Fables programming language and the modeling environment is under constant development. There're new builds available weekly to update your current installation, but you can also downgrade back to any installed version of the IME and the compiler at anytime. In the following we describe how to perform these tasks.

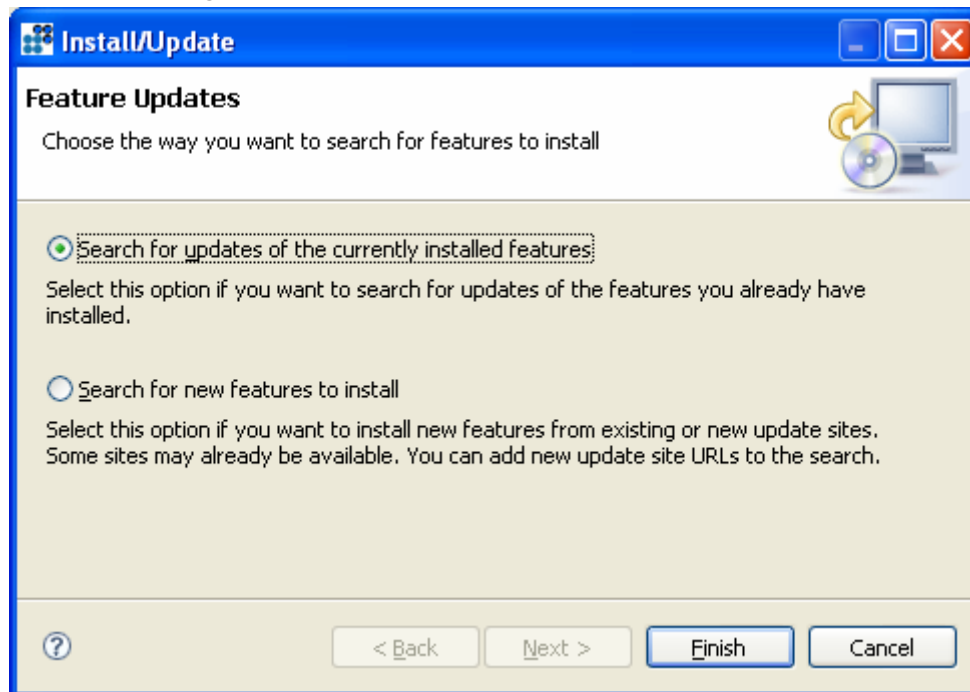
8.1 Upgrading the IME

This task can be performed with a few clicks.

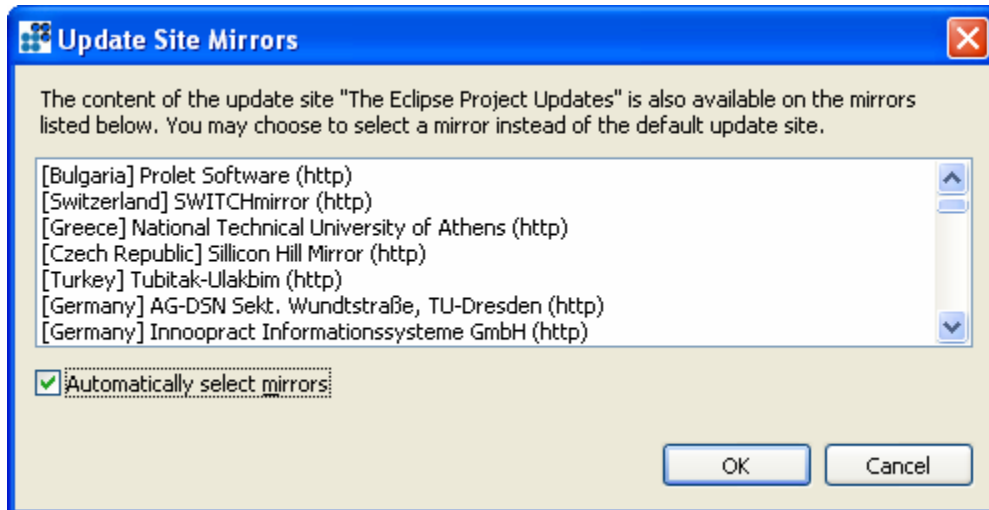
1. Start the Fables IME
2. Click on Help → Software Updates → Find and Install...



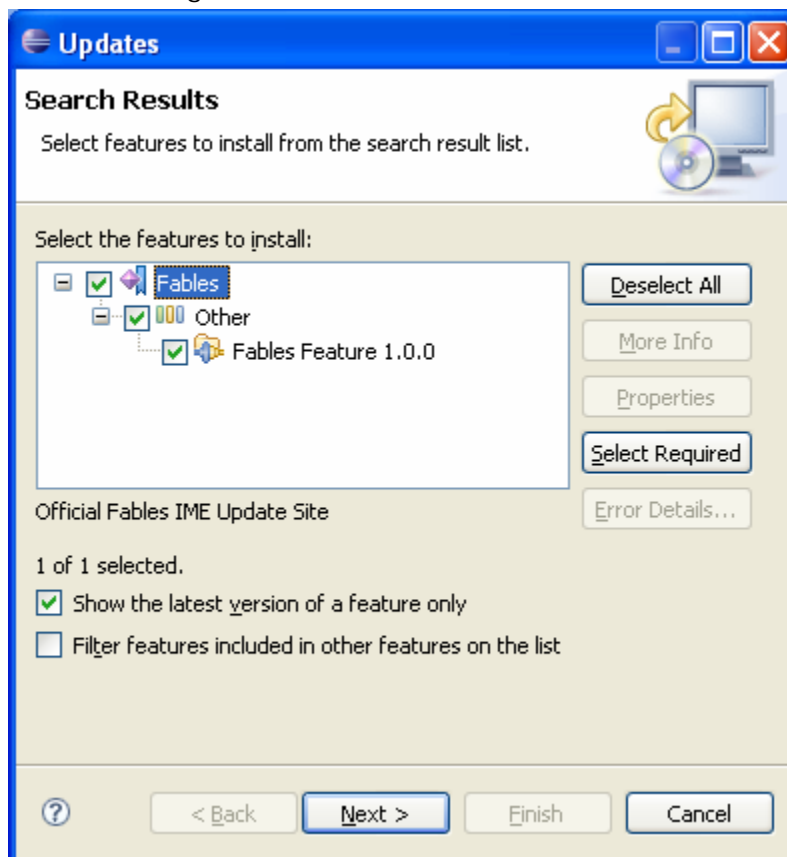
3. Select the "Search for updates of the currently installed features" option on the next dialog, then click Finish.



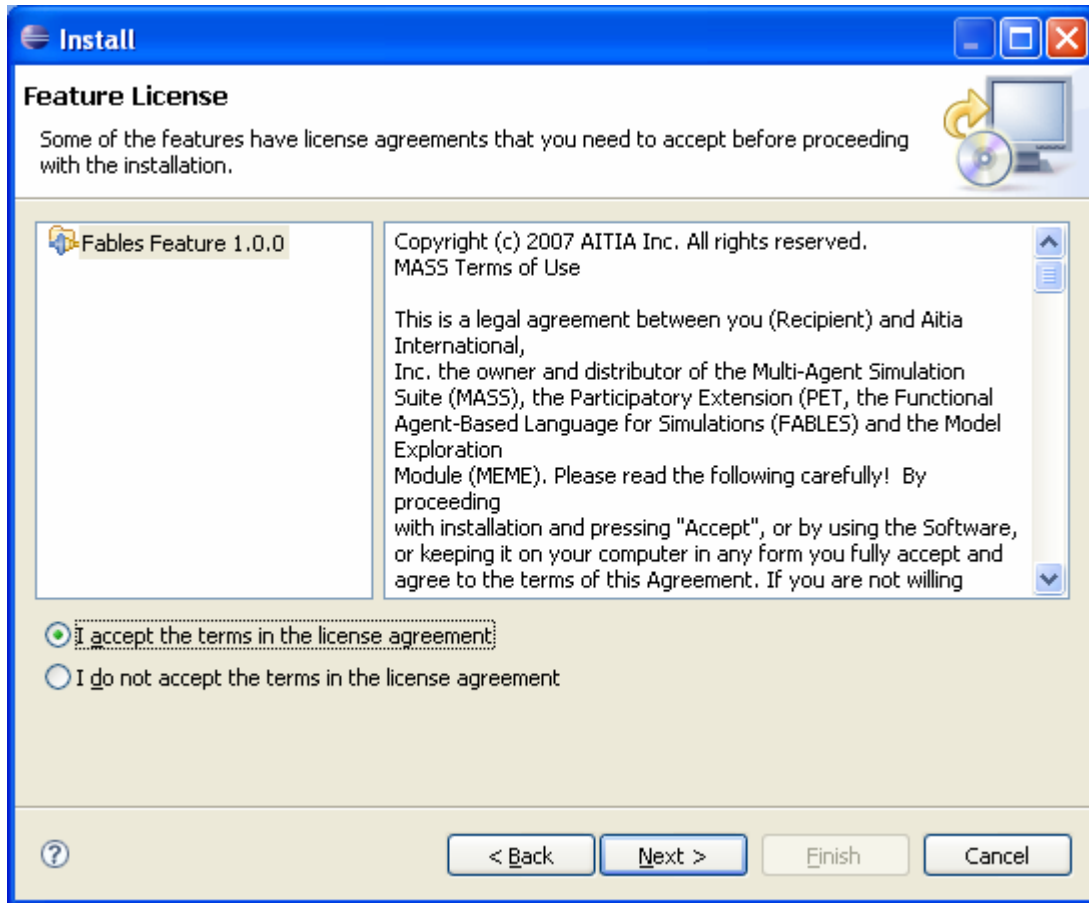
- The IME updates not only the compiler and the IME, but every Eclipse component it uses. There are several update mirror sites available, you'll be asked to choose one of them. Selecting the option "Automatically select mirrors" is just fine.



- The update manager is going to contact all of the update sites if there's an update available for any of the installed components. If there are available updates you'll be asked to confirm downloading. We suggest updating every component to the latest version. If there are no available updates the updater will return the dialog "No features available". When finished, click on Next.



6. On the next dialog you'll be asked to accept the Fables license agreement. Select the corresponding button and click Next.



7. In the final dialog you are asked to select the installation directory for the updated Fables plug-in. The default installation folder is fine, simply click on Finish.
8. The new version of the plug-in is going to be downloaded. After the download you'll be asked to install it. Simply click on the Install All button.



9. You'll be asked to restart the IME, because some of the changes need a restart to take effect. Click on Yes to restart.

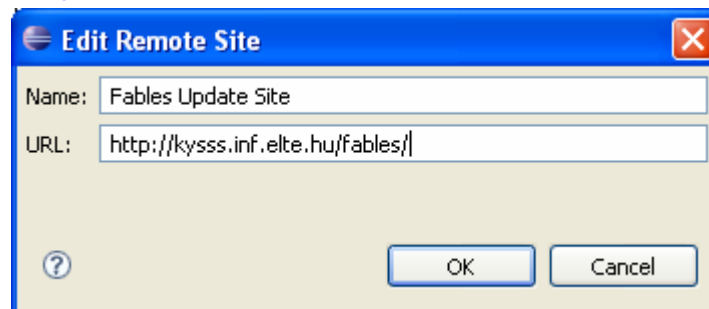
10. The update process is finished!

The IME core is now updated. You should also download and unpack the Runtime Libraries from the <http://kysss.inf.elte.hu/fables> site to the default Eclipse installation directory. This package contains several additional features, like new sample models, runtime libraries, documentation, etc.

8.2 Integrating the Fables plug-in to an existing Eclipse installation

It is possible to install the Fables plug-in into an existing Eclipse installation

1. Click on Help → Software Updates → Find and Install...
2. Select the "Search for new features to install" option and click on Next.
3. On the next "Update sites to visit" dialog click on the "New Remote Site" button, and set the name field to "Fables Update Site" and fill in the URL field with <http://kysss.inf.elte.hu/fables/>.



4. Make the newly added update site active, and click on Finish.
5. The Eclipse Updater guides you through the installation procedure of the new plug-in.

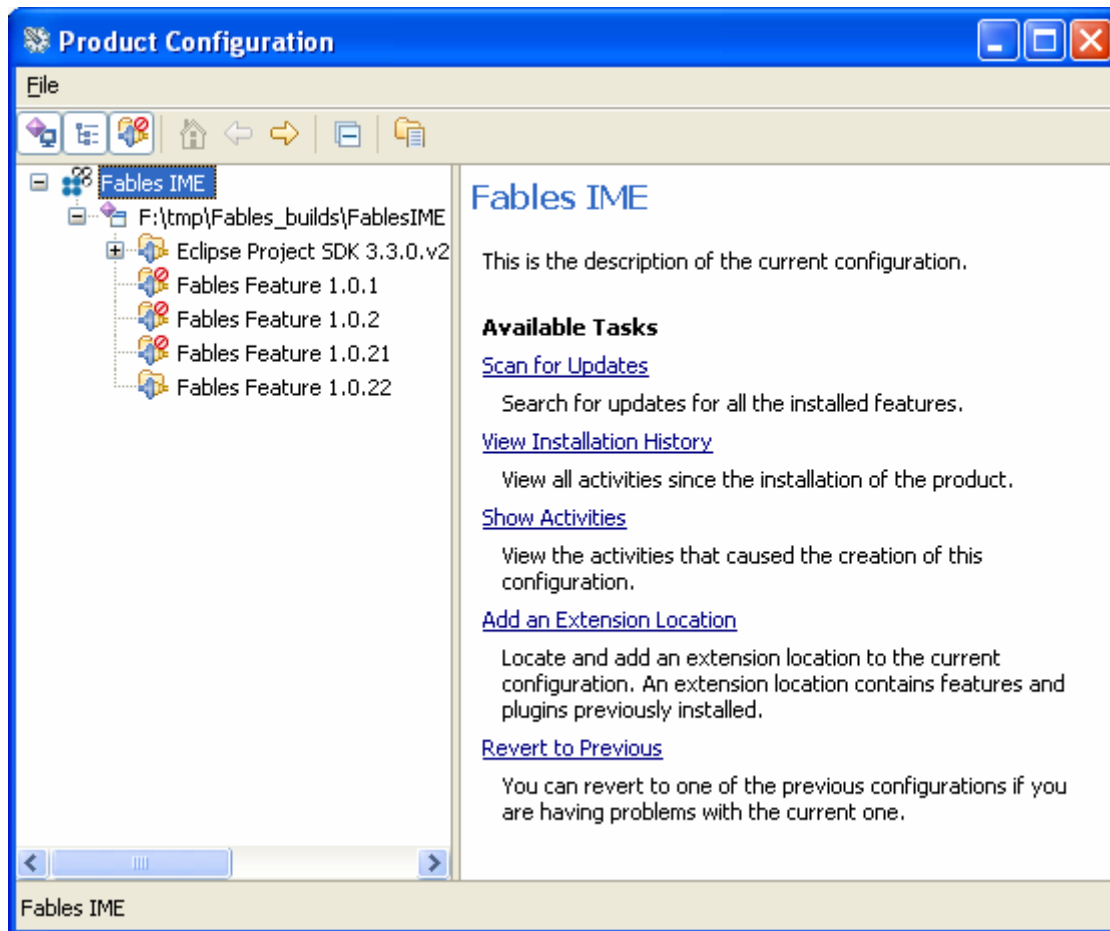
Your Eclipse is now updated. You should also download and unpack the Runtime Libraries from the <http://kysss.inf.elte.hu/fables> site to the default Eclipse installation directory. This package contains several additional features, like the

sample models, the runtime libraries, some documentation, etc. This package has to be unpacked for the Fables plug-in to work properly under Eclipse.

8.3 Disabling, uninstalling and switching between plug-in versions

If you have more than one version of Fables plug-ins installed, you can switch between them by selecting Help → Software Updates → Manage Configuration.

A dialog shows up where you can disable, switch or uninstall the various plug-in versions.



Except the base version of the Fables Feature (v.1.0.0) all of the Features can be uninstalled. The red mark shows that the plug-in is currently disabled (because there's a newer version installed).

If you've installed the Fables plug-in into an existing Eclipse installation, you'll be able to remove all of the Fables Features.

Left-clicking any of the features displays more information about the plug-in.

Right-clicking any of the features displays a context menu.

9 Appendix

9.1 The Fables EBNF grammar

```

NON-TERMINALS
Start ::= ( Import )* Model
Import ::= <IMPORT> <IDENTIFIER> ( <SEMICOLON> )+
Model ::= ( <DOC_COMMENT> )?
        <MODEL> ModelName <LBRACE>
        ( Definition | ClassDefinition | Schedule | StartUp )*
        <RBRACE> ( <SEMICOLON> )? <EOF>

ModelName ::= Name
Definition ::= ConstantDefinition | VariableDefinition
VariableDefinition ::= ( <DOC_COMMENT> )?
                    <VAR> Name <SEMICOLON>
ConstantDefinition ::= ( <DOC_COMMENT> )?
                    Name
                    ("()" | <LPAREN> <RPAREN> | ArgumentList <RPAREN> )?
                    "=" CompleteExpression <SEMICOLON>
LocalDefinition ::= Name ( <LPAREN> ArgumentList <RPAREN> )?
                  "=" CompleteExpression
ClassDefinition ::= ( <DOC_COMMENT> )?
                  <CLASS> ClassName <LBRACE>
                  ( Definition | Schedule | ClassStartUp )*
                  <RBRACE> ( <SEMICOLON> )?
ClassStartUp ::= ( <DOC_COMMENT> )?
               <STARTUP> ( "(" | <LPAREN> <RPAREN> )? "{"
               ( <SEMICOLON> | Expression <SEMICOLON> )* "}" ( <SEMICOLON>
               )?
StartUp ::= ( <DOC_COMMENT> )?
           <STARTUP> ( ( "(" ParameterList ")" ) | "(" )? "{"
           ( <SEMICOLON> | Expression <SEMICOLON> )* "}"
           ( <SEMICOLON> )?
ParameterList ::= Identifier ( "," Identifier )*
Schedule ::= ( <DOC_COMMENT> )?
           <SCHEDULE> ( ScheduleName )? ( <CYCLIC> Integer )? "{"
           ( Event )*
           "}" ( <SEMICOLON> )?
ScheduleName ::= Name
Event ::= Time ":" ( <SEMICOLON> )* Expression
        ( <SEMICOLON> Expression )* <SEMICOLON>
Time ::= Integer
ClassName ::= Name
ArgumentList ::= Variable ( "," Variable )*
Variable ::= Name
CompleteExpression ::= Expression
                   ( <WHERE> <LPAREN> LocalDefinition ( "," LocalDefinition )* <RPAREN> )?
AtomExpression ::= Identifier ( "(" )?
                | Number
                | String
BasicExpression ::= AtomExpression

```

```

    |      SetExpression
    |      ListExpression
    |      BracketedExpression
    |      ForEachExpression
BracketedExpression ::= <LPAREN>( TupleExpression | BinaryOperator )<RPAREN>
ForEachExpression  ::= <FOR> <EACH> Name <IN> Expression <DO> Expression
TupleExpression ::= Expression ( "," Expression )*
BinaryOperator ::= MultiplicativeOperator
    |      AdditiveOperator
    |      LogicalOperator
    |      RelationalOperator
    |      InOperator
LogicalOperator ::= AndOperator
    |      OrOperator
Member ::= Name
    |      Integer
FunctionExpression ::= ( <DOC_COMMENT> )?
                    BasicExpression ( ( MemberOperator Member | BasicExpression ) )*
MemberOperator ::= "."
UnaryExpression ::= ( FunctionExpression | UnaryMinusExpression | KhiOperator )
UnaryMinusExpression ::= "-" FunctionExpression
PowerExpression ::= UnaryExpression ( "^" UnaryExpression )*
MultiplicativeExpression ::= PowerExpression
                        ( MultiplicativeOperatorPowerExpression )*
MultiplicativeOperator ::= MulOperator
    |      PerOperator
    |      DivOperator
    |      ModOperator
    |      RemOperator
    |      SetMinusOperator
AdditiveExpression ::= MultiplicativeExpression
                    ( AdditiveOperator MultiplicativeExpression )*
AdditiveOperator ::= PlusOperator | MinusOperator
ConcatenateExpression ::= AdditiveExpression ( "++" AdditiveExpression )*
RelationalExpression ::= ConcatenateExpression ( ( EqualExpression | NotEqualExpression |
LesserExpression | LesserOrEqualExpression | GreaterExpression | GreaterOrEqualExpression | InOperator
ConcatenateExpression ) )?
EqualExpression ::= EqualOperator ConcatenateExpression
                    ( EqualOperator ConcatenateExpression )*
NotEqualExpression ::= NotEqualOperator ConcatenateExpression
                    ( NotEqualOperator ConcatenateExpression )*
LesserExpression ::= LesserOperator ConcatenateExpression
                    ( LesserOperator ConcatenateExpression )*
LesserOrEqualExpression ::= LesserOrEqualOperator ConcatenateExpression
                    ( LesserOrEqualOperator ConcatenateExpression )*
GreaterExpression ::= GreaterOperator ConcatenateExpression
                    ( GreaterOperator ConcatenateExpression )*
GreaterOrEqualExpression ::= GreaterOrEqualOperator ConcatenateExpression
                    ( GreaterOrEqualOperator ConcatenateExpression )*
RelationalOperator ::= EqualOperator
    |      NotEqualOperator

```

```

| LesserOperator
| GreaterOperator
| LesserOrEqualOperator
| GreaterOrEqualOperator
NotExpression ::= RelationalExpression | UnaryNotExpression
UnaryNotExpression ::= <NOT> RelationalExpression
LogicalExpression ::= NotExpression
                    ( ( LogicalAndExpression | LogicalOrExpression ) ) *
LogicalAndExpression ::= AndOperator NotExpression
LogicalOrExpression ::= OrOperator NotExpression
NonImperativeExpression ::= LogicalExpression
AssignmentExpression ::= NonImperativeExpression
                       ( ":" = " ImperativeExpression ) ?
ConstructExpression ::= <NEW> Identifier
                    ( ( "[" ( "]" | Identifier ":" = " Expression ( "," Identifier ":" = " Expression ) * "]" ) ) ) ?
DestructExpression ::= <DELETE> FunctionExpression
ImperativeExpression ::= AssignmentExpression
| ConstructExpression
| DestructExpression
ConditionalExpression ::= ImperativeExpression
                      ( ( "=" > " ImperativeExpression
                        ( "|" ImperativeExpression "=" > " ImperativeExpression ) *
                        ( <OTHERWISE> ( "=" > " ) ? ImperativeExpression ) ? ) ) ?
Expression ::= ConditionalExpression
SetExpression ::= <LBRACE> (
                ( ( Iteration ( IterationCondition ) ? )
                  | ( Expression ( IntervalExpression
                                | ( ( MultiValueExpression ) ? ( Iterations ( IterationCondition ) ? ) ? ) ) ) )
                ) ?
                <RBRACE>
ListExpression ::= <LBRACKET> (
                ( ( Iteration ( IterationCondition ) ? )
                  | ( Expression ( IntervalExpression
                                | ( ( MultiValueExpression ) ? ( Iterations ( IterationCondition ) ? ) ? ) ) ) )
                ) ? <RBRACKET>
MultiValueExpression ::= "," Expression ( "," Expression ) *
IntervalExpression ::= ".." Expression ( "," Expression ".." Expression ) *
                    ( "|" Expression ) ?
Iterations ::= ":" Iteration ( "," Iteration ) *
IterationCondition ::= <WHEN> Expression
Iteration ::= Name <IS> Expression
Number ::= Integer | Float
PlusOperator ::= "+"
KhiOperator ::= <KHI> <LPAREN> Iteration ( "," Iteration ) *
              ( IterationCondition ) ? <RPAREN>
MinusOperator ::= "-"
MulOperator ::= "*"
PerOperator ::= "/"
DivOperator ::= <DIV>
ModOperator ::= <MOD>
RemOperator ::= <REM>

```

```
SetMinusOperator ::= "\\\"
PowOperator      ::= "^\"
AndOperator      ::= "<AND>\"
OrOperator       ::= "<OR>\"
InOperator       ::= "<IN>\"
EqualOperator    ::= "\"==\"
NotEqualOperator ::= "\"<>\"
LesserOperator   ::= "\"<\"
GreaterOperator  ::= "\">\"
LesserOrEqualOperator ::= "\"<=\"
GreaterOrEqualOperator ::= "\">=\"
Name             ::= "<IDENTIFIER>\"
Identifier       ::= "( (<IDENTIFIER> | <SELF> ) )\"
Integer ::= <INTEGER_LITERAL>\"
Float           ::= <FLOATING_POINT_LITERAL>\"
String         ::= <STRING_LITERAL>
```