Sommaire

I. Présentation 3

A. Modèle Mvc 3
B. Historique 3

II. Installation 4

A. Installer Ruby sous Windows 4
B. Connecteurs 6
C. Accès aux base 6

III. Les éditeurs 7

A. Scite 7
B. Netbeans 7
C. Installation 8
D. Utilisation 11

IV. Le langage 15

A. Hello the World 15
B. Paramètres de la ligne de Commande 15
C. Les chaînes 15
1. Here Doc 15
2. Extraction 15
3. Quelques fonctions 15
4. Interpolation 15
5. Expression régulière 15
D. Les fonctions 16
E. Variables 16
1. Nommage 16
2. Préfixe 16
F. Tableaux 16
G. Date 16
H. Structure du contrôle 16
1. Alternative 16
2. Répétitives 17
3. Ecritures simplifiées 17
4. expression ternaire 17
5. Case 17
6. Gestion de blocs 17
7. Itération 17
8. Gestion d'erreurs 17
9. Les modules 18

V. Les objets 19

A. Classe statique 19
B. Création d'une classe 19
C. La portée 19
D. Création d'un objet 19
E. Héritage 19
F. Utilisation d'une classe héritée 20
G. Lire une propriété 20
H. Classe statique 20
I. connexion à MySql 20
1. La classe 20
2. Le code 21

VI. Rails 22

A. Installation de WebBricks 22
B. Structure de Rails 22
C. Création d'un contrôleur 22
D. La vue 22
E. Configuration de la base 23
F. création des databases et des tables 24
1. Création des bases 24
2. Création du script de crétaion de tables 24
3. Modification de table 25
4. Type de champ 25
5. Création des tables 25
G. Création d'un contrôleur 25
H. La vue 25
I. Les lookups 26
J. La gestion des formulaires 26
1. La page text.html 26
2. Le contrôleur 26
3. La vue 26
4. Le formulaire Ruby 26
K. La feuille de style 27
L. Conversion 27
M. Envoi de mail 27

VII. Ressources 28

I. Présentation

A. Modèle Mvc

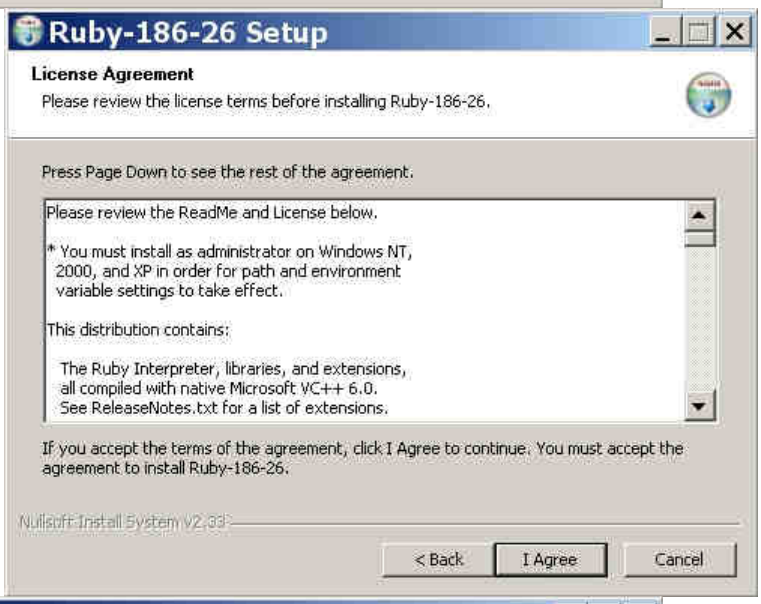
- Les modèles qui sont les classes qui assurent la gestion des données : la structure de ces classes étant déterminer par Rails à partir de la base de donnée
- Les vues qui déterminent l'affichage des informations : généralement une combinaison de html et de ruby .rhtml
- Les contrôleurs qui réagissent aux actions de l'utilisateur et qui répondent généralement à travers la vue : on peut créer facilement des squelettes de contrôleurs pour les actions simples. Ruby on Rails contraint le développeur à utiliser une arborescence qui sépare ces trois composantes, il y a donc obligatoirement les répertoires « Model », « View », « Controller » dans chaque projet Rails.

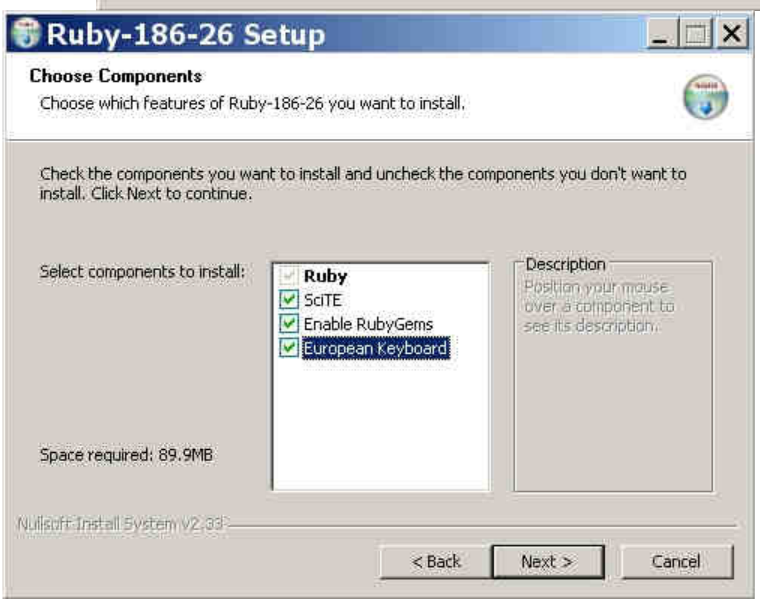
B. Historique

1995 Création de Ruby par Matsumoto
2001 Création de Rubygems par Ryan Leavengood
2004 Création de RoR

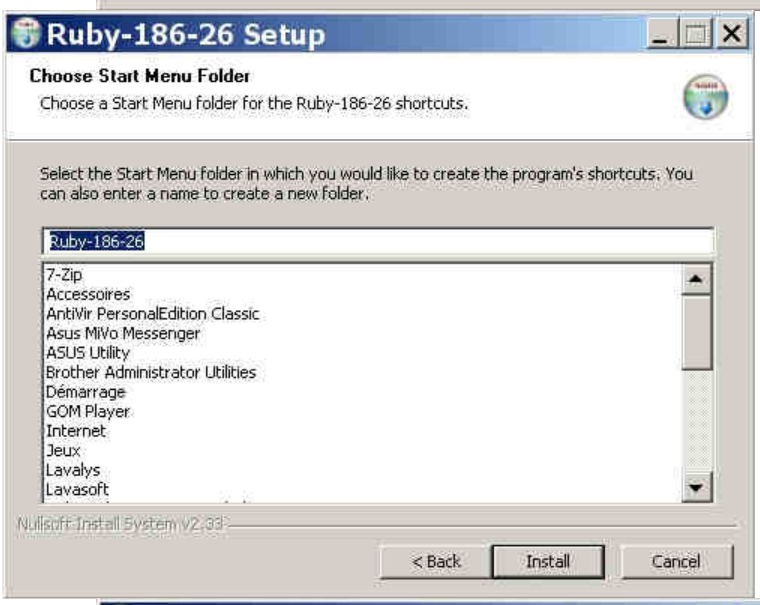
II. Installation

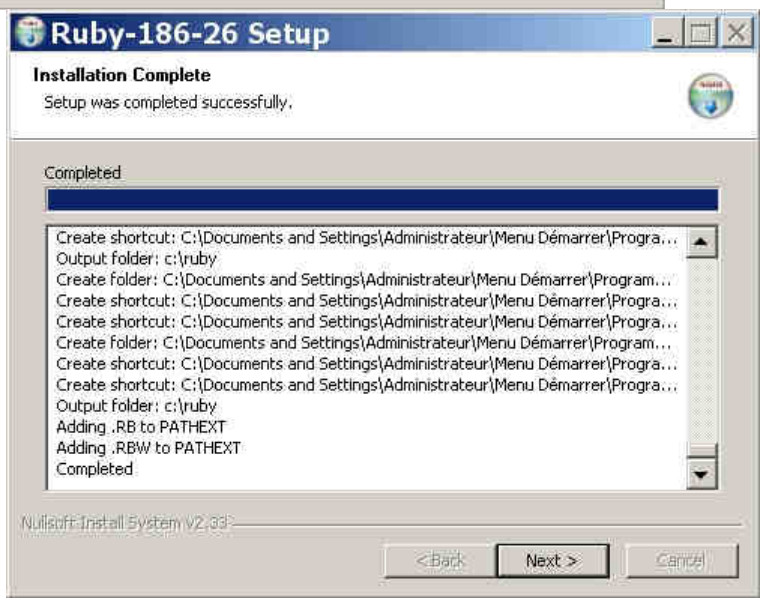
A. Installer Ruby sous Windows






Ruby-186-26 Setup
Choose Install Location
Choose the folder in which to install Ruby-186-26.
Setup will install Ruby-186-26 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
c:\ruby
Browse...
Space required: 89.9MB
Space available: 27.7GB
Nullsoft Install System v2.33
< Back
Next >
Cancel


Ruby-186-26 Setup
Choose Start Menu Folder
Choose a Start Menu folder for the Ruby-186-26 shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
Ruby-186-26
7-Zip
Accessoires
AntiVir PersonalEdition Classic
Asus MiVo Messenger
ASUS Utility
Brother Administrator Utilities
Démarrage
GOM Player
Internet
Jeux
Lavalys
Lavasoft
Nullsoft Install System v2.33
< Back
Install
Cancel


Ruby-186-26 Setup
Installation Complete
Setup was completed successfully.
Completed
Create shortcut: C:\Documents and Settings\Administrateur\Menu Démarrer\Progra...
Output folder: c:\ruby
Create folder: C:\Documents and Settings\Administrateur\Menu Démarrer\Program...
Create shortcut: C:\Documents and Settings\Administrateur\Menu Démarrer\Progra...
Create shortcut: C:\Documents and Settings\Administrateur\Menu Démarrer\Progra...
Create folder: C:\Documents and Settings\Administrateur\Menu Démarrer\Program...
Create shortcut: C:\Documents and Settings\Administrateur\Menu Démarrer\Progra...
Create shortcut: C:\Documents and Settings\Administrateur\Menu Démarrer\Progra...
Output folder: c:\ruby
Adding .RB to PATHEXT
Adding .RBW to PATHEXT
Completed
Nullsoft Install System v2.33
< Back
Next >
Cancel

B. Connecteurs

DB2 http://raa.ruby-lang.org/project/ruby-db2
MySQL http://www.tmtm.org/en/mysql/ruby
Oracle http://rubyforge.org/projects/ruby-oci8
Postgres http://ruby.scripting.ca/postgres/
SQLite http://rubyforge.org/projects/sqlite-ruby
Postgresql from the Ruby-DBI http://rubyforge.org/projects/ruby-dbi

C. Accès aux base

gem search mysql --remote
gem install mysql
gem install ruby-postgres

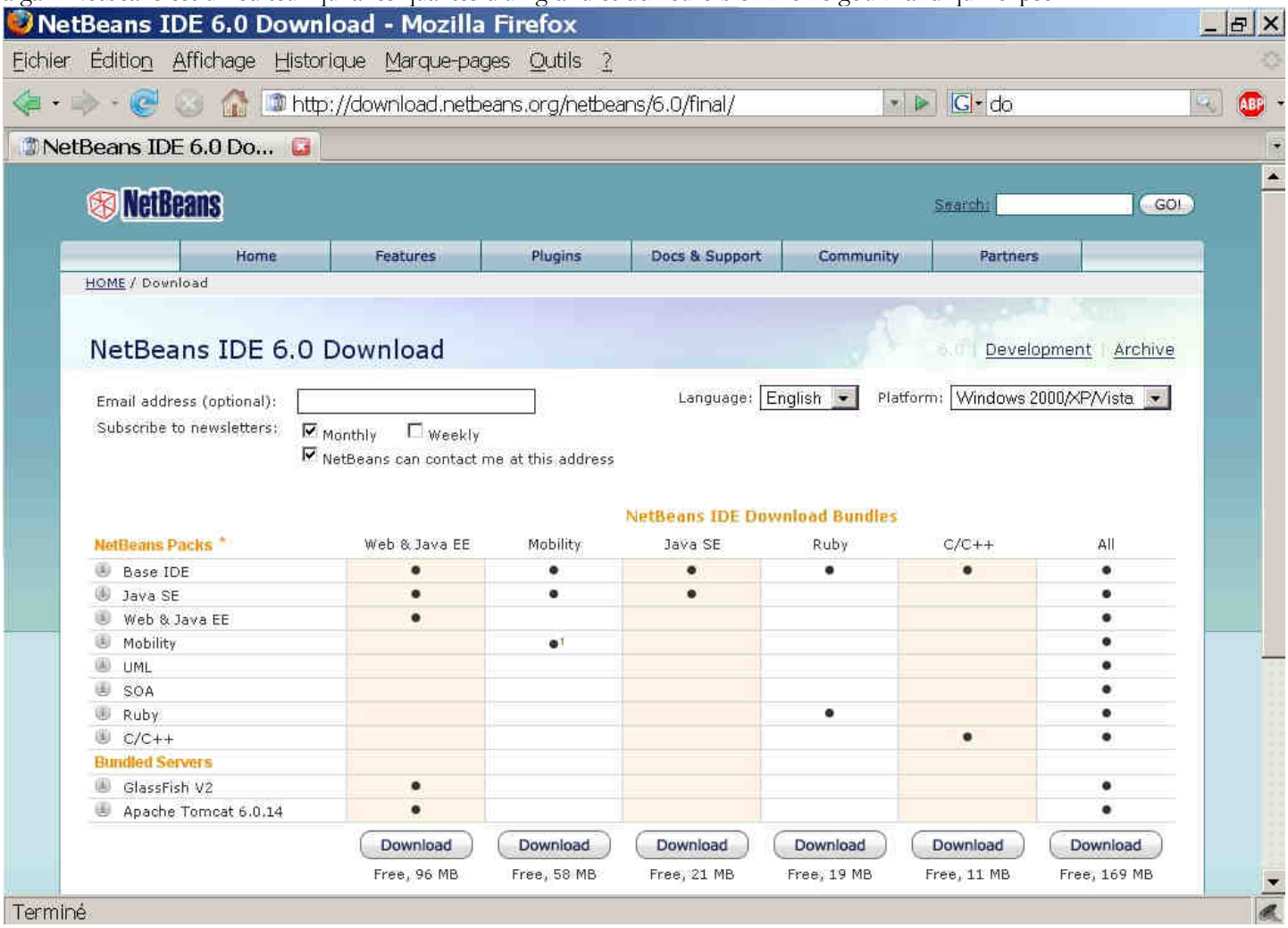
III. Les éditeurs

A. Scite

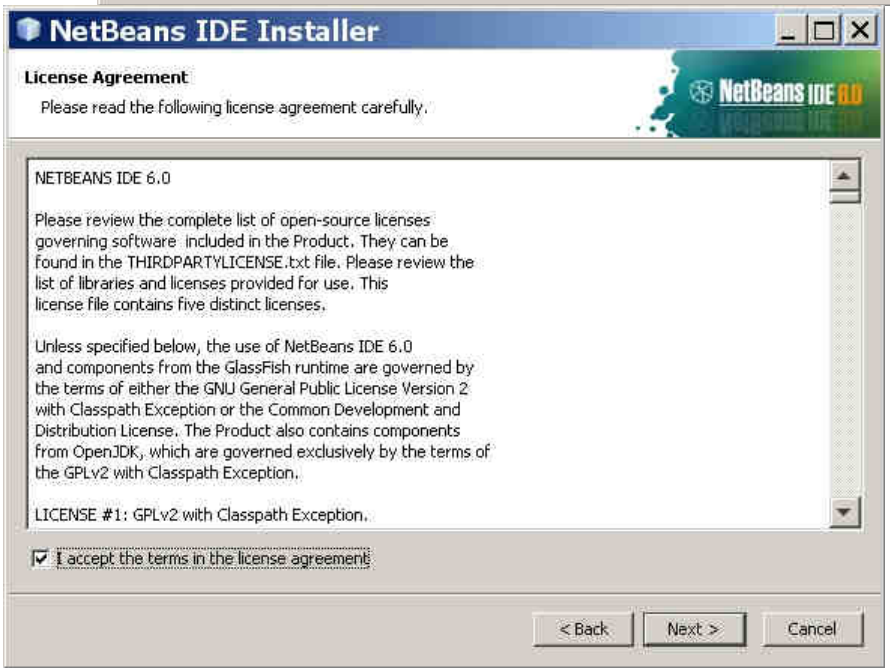
Lors de l'installation de Ruby sous Windows, l'éditeur Gpl Scite qui utilise la coloration syntaxique est installé sur votre système. L'outil est intéressant. Il n'offre pas les caractéristiques d'un véritable IDE.

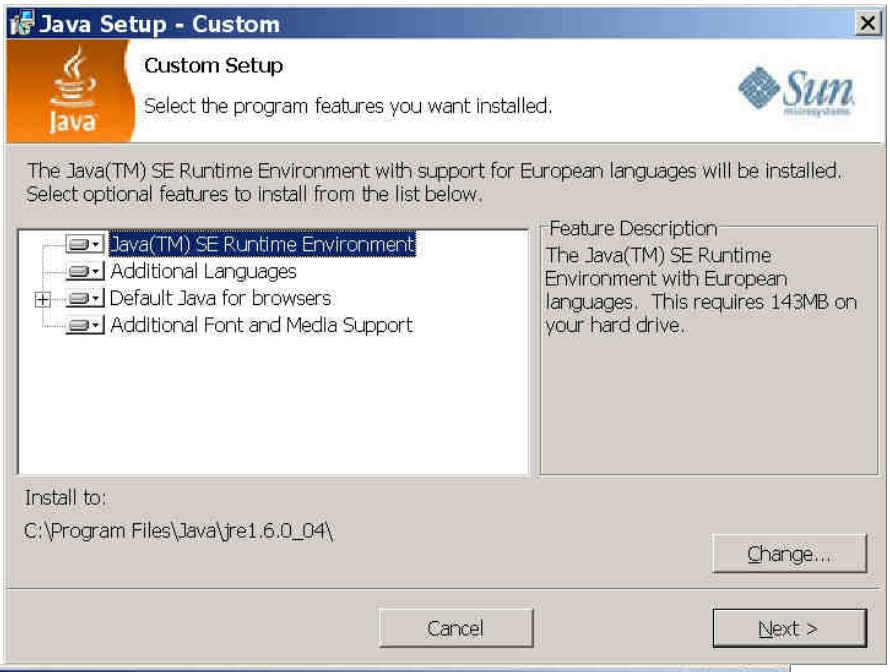
B. Netbeans

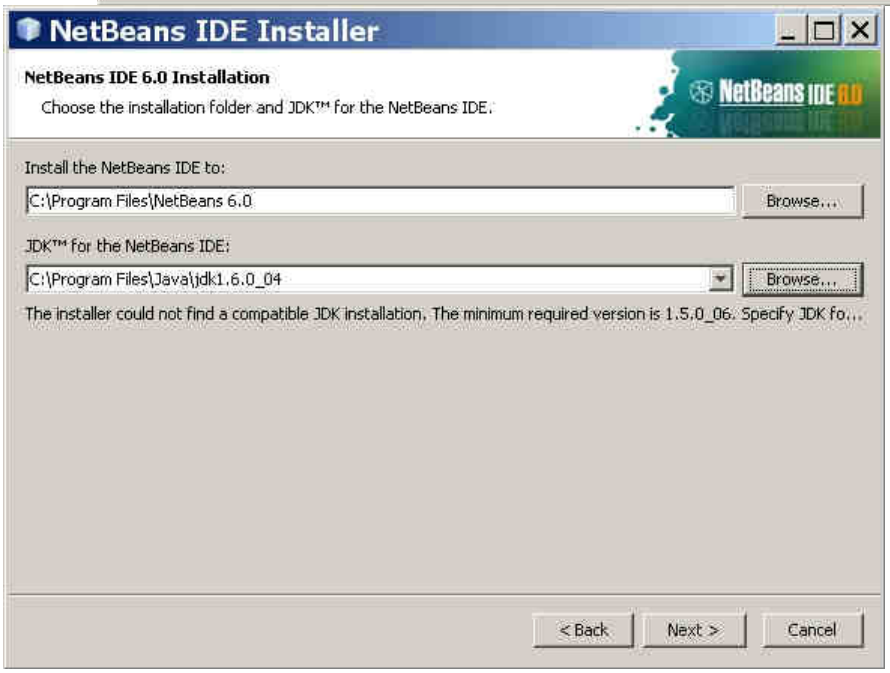
Eclipse est en passe de devenir un standard dans le monde du développement. De mon point de vue, il est une véritable usine à gaz. Netbeans est un éditeur qui a les qualités d'un grand et demeure bien moins gourmand qu'Eclipse.

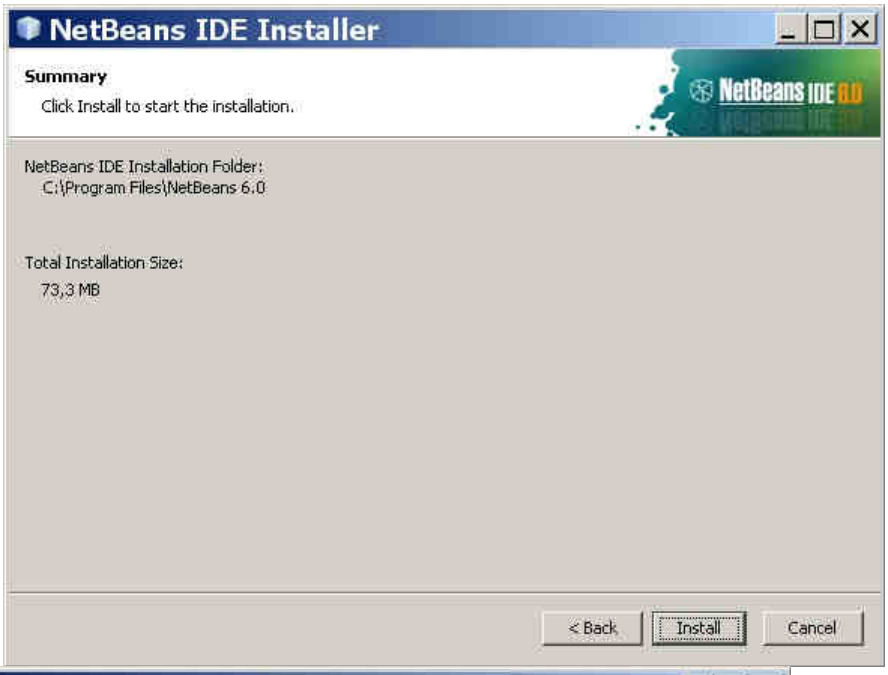


C. Installation



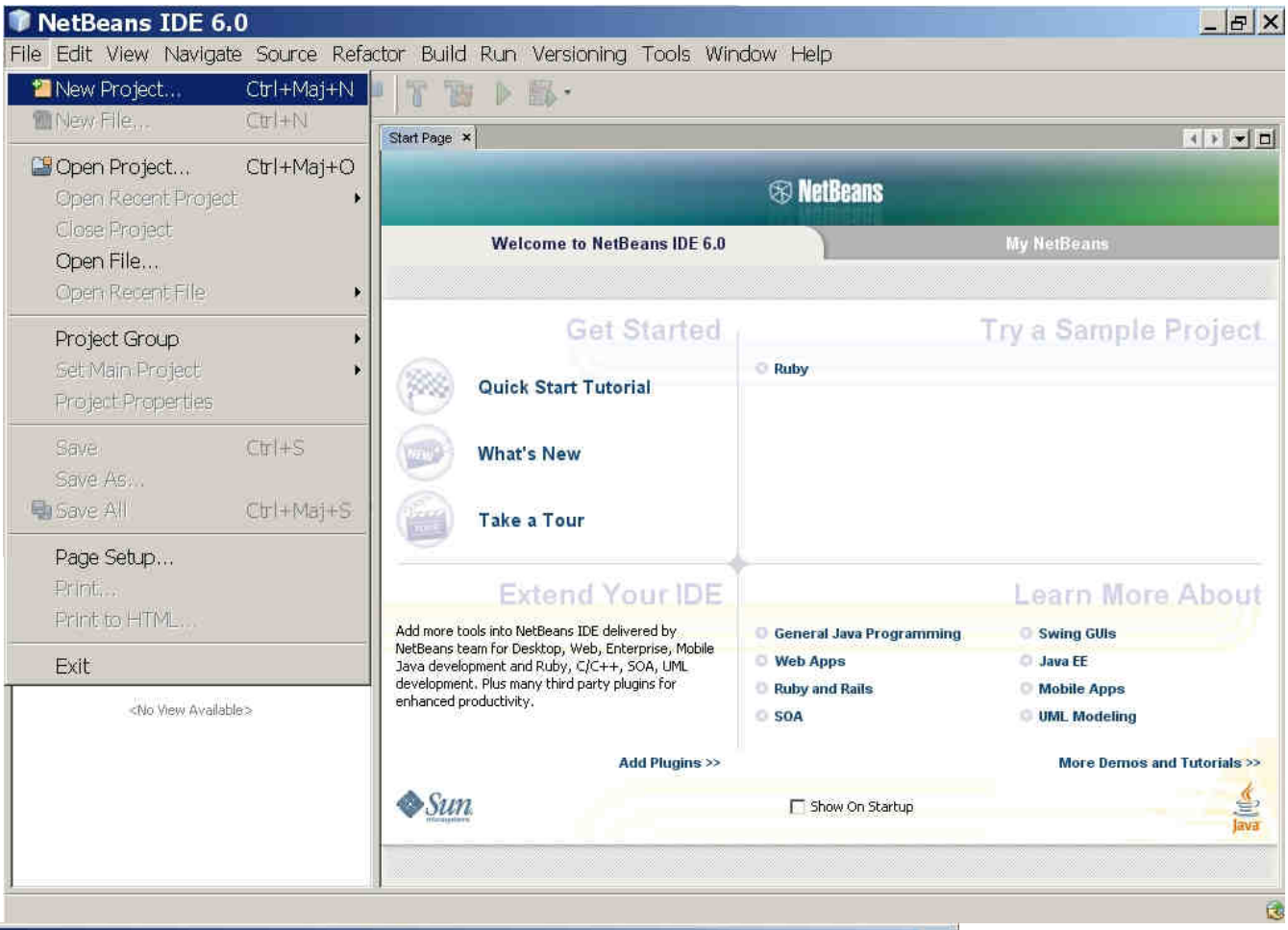

Java Setup - Custom
Custom Setup
Select the program features you want installed.
Sun
The Java(TM) SE Runtime Environment with support for European languages will be installed.
Select optional features to install from the list below.
Java(TM) SE Runtime Environment
Additional Languages
Default Java for browsers
Additional Font and Media Support
Feature Description
The Java(TM) SE Runtime Environment with European languages. This requires 143MB on your hard drive.
Install to:
C:\Program Files\Java\jre1.6.0_04\
Change...
Cancel
Next >

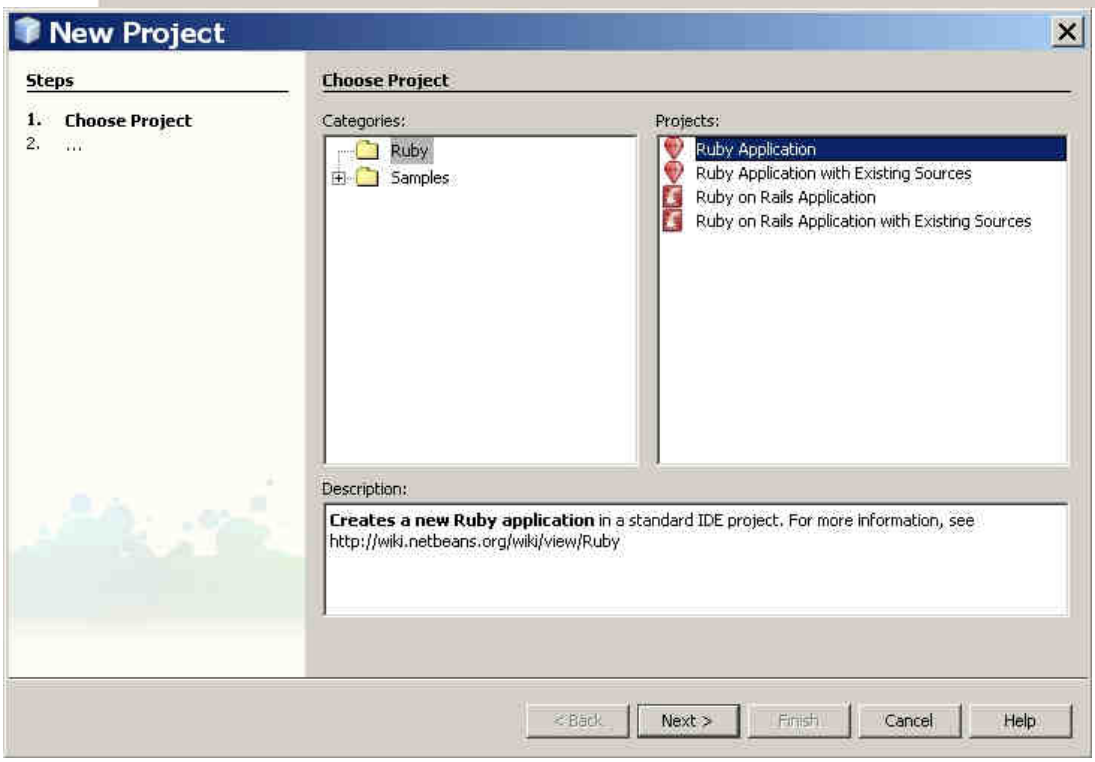

NetBeans IDE Installer
NetBeans IDE 6.0 Installation
Choose the installation folder and JDK™ for the NetBeans IDE.
NetBeans IDE 6.0
Install the NetBeans IDE to:
C:\Program Files\NetBeans 6.0
Browse...
JDK™ for the NetBeans IDE:
C:\Program Files\Java\jdk1.6.0_04
Browse...
The installer could not find a compatible JDK installation. The minimum required version is 1.5.0_06. Specify JDK fo...
< Back
Next >
Cancel

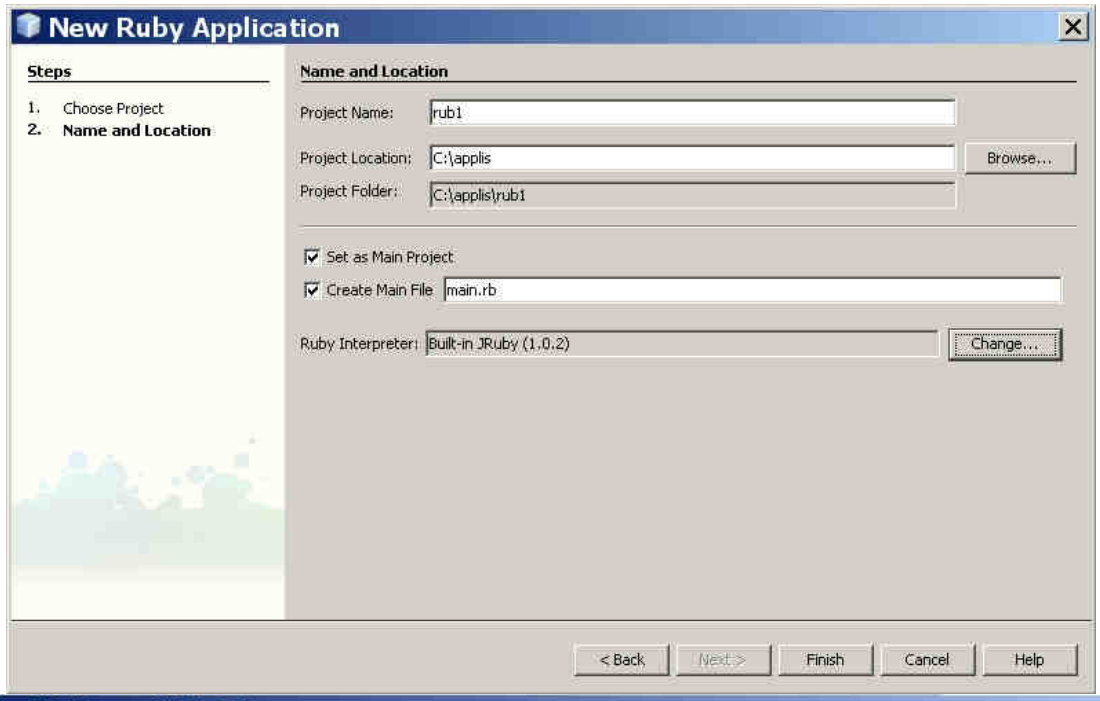

NetBeans IDE Installer
Summary
Click Install to start the installation.
NetBeans IDE 6.0
NetBeans IDE Installation Folder:
C:\Program Files\NetBeans 6.0
Total Installation Size:
73,3 MB
< Back
Install
Cancel

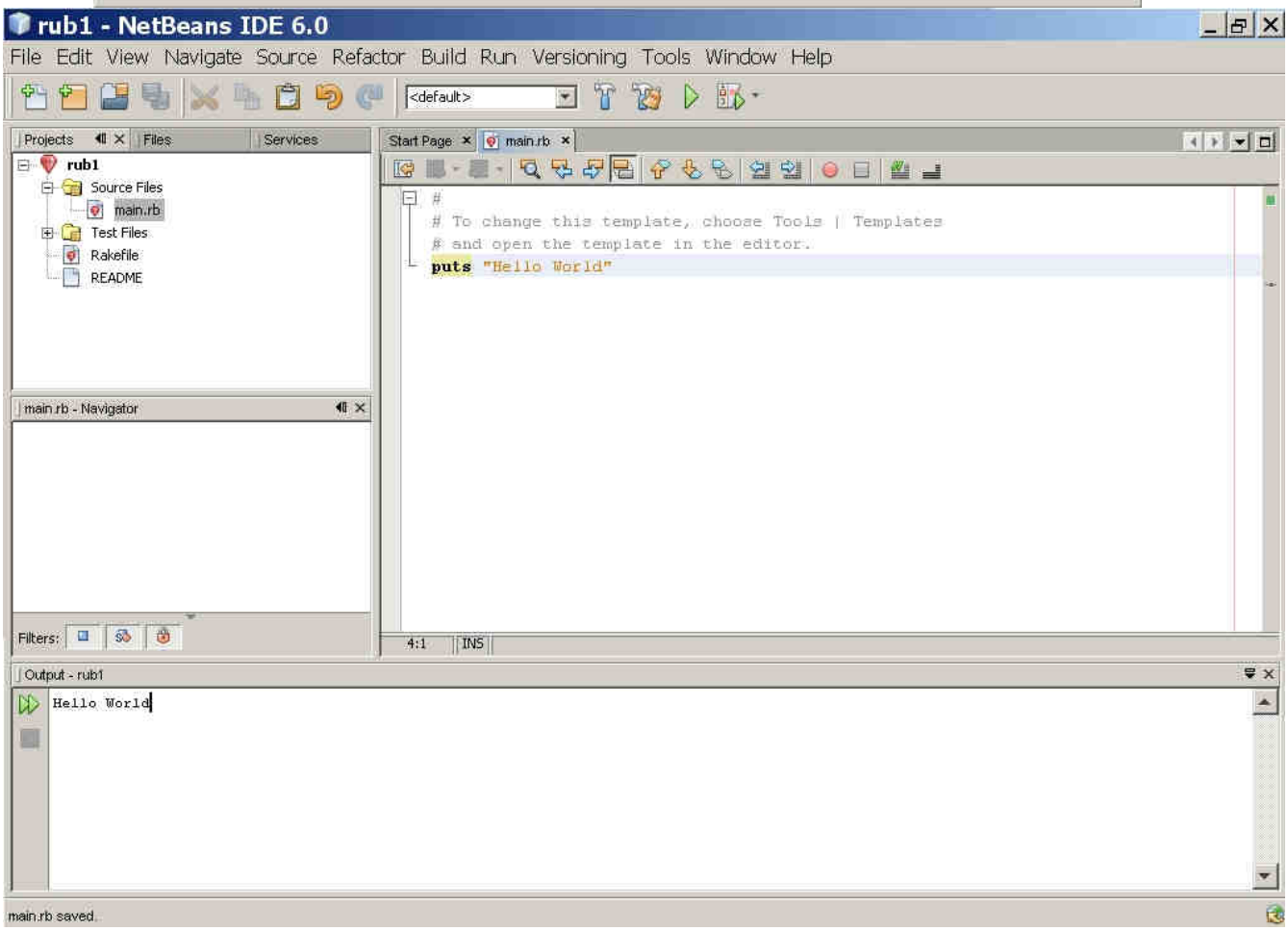

NetBeans IDE Installer
Setup Complete
Click Finish to finish the NetBeans IDE setup.
NetBeans IDE 6.0
Installation completed successfully.
To launch the IDE, use either the Start menu or the NetBeans desktop icon.
To change installed components and add NetBeans plugins, use Plugin Manager that is an integral part of NetBeans IDE.
Finish

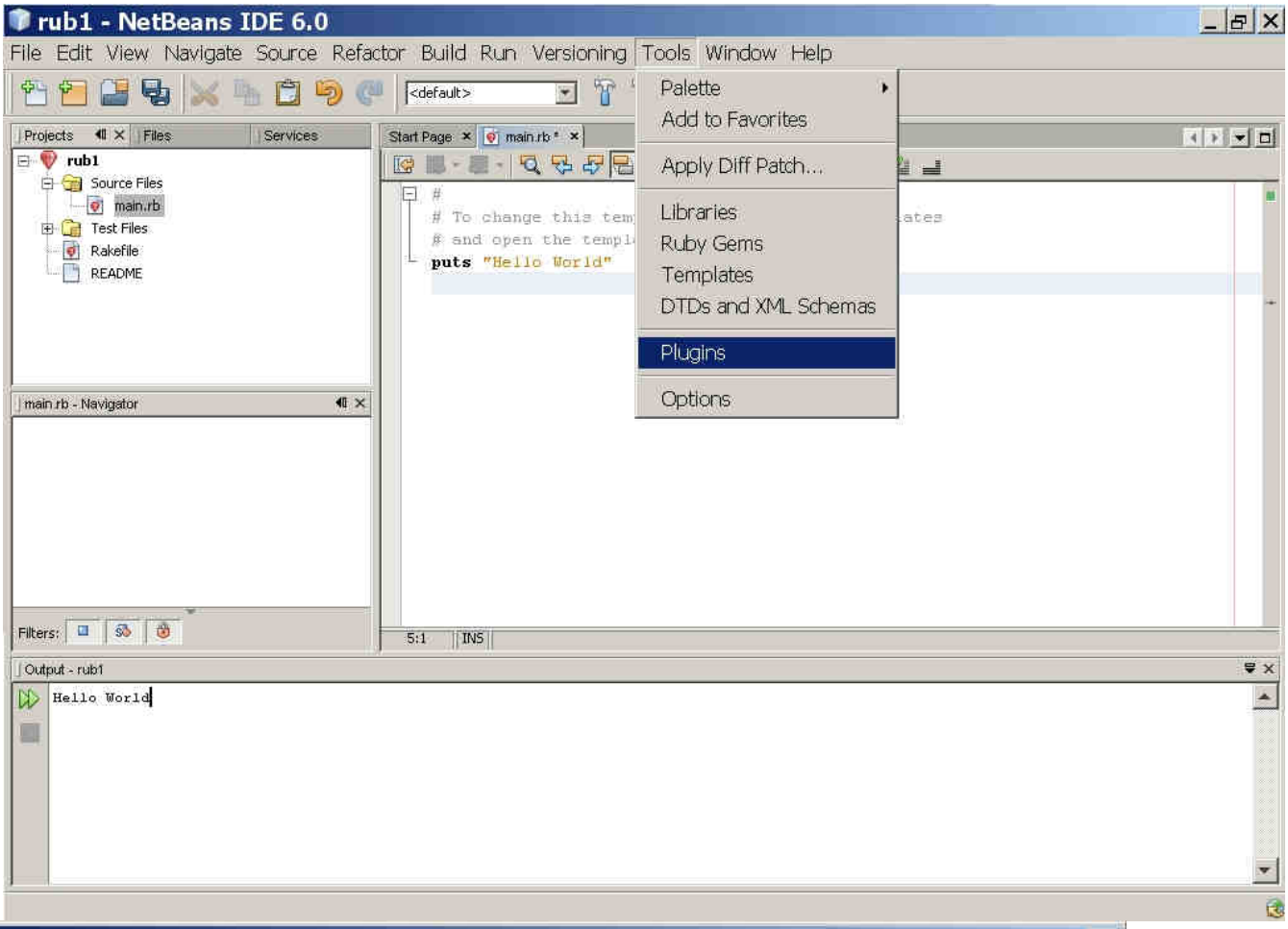
D. Utilisation

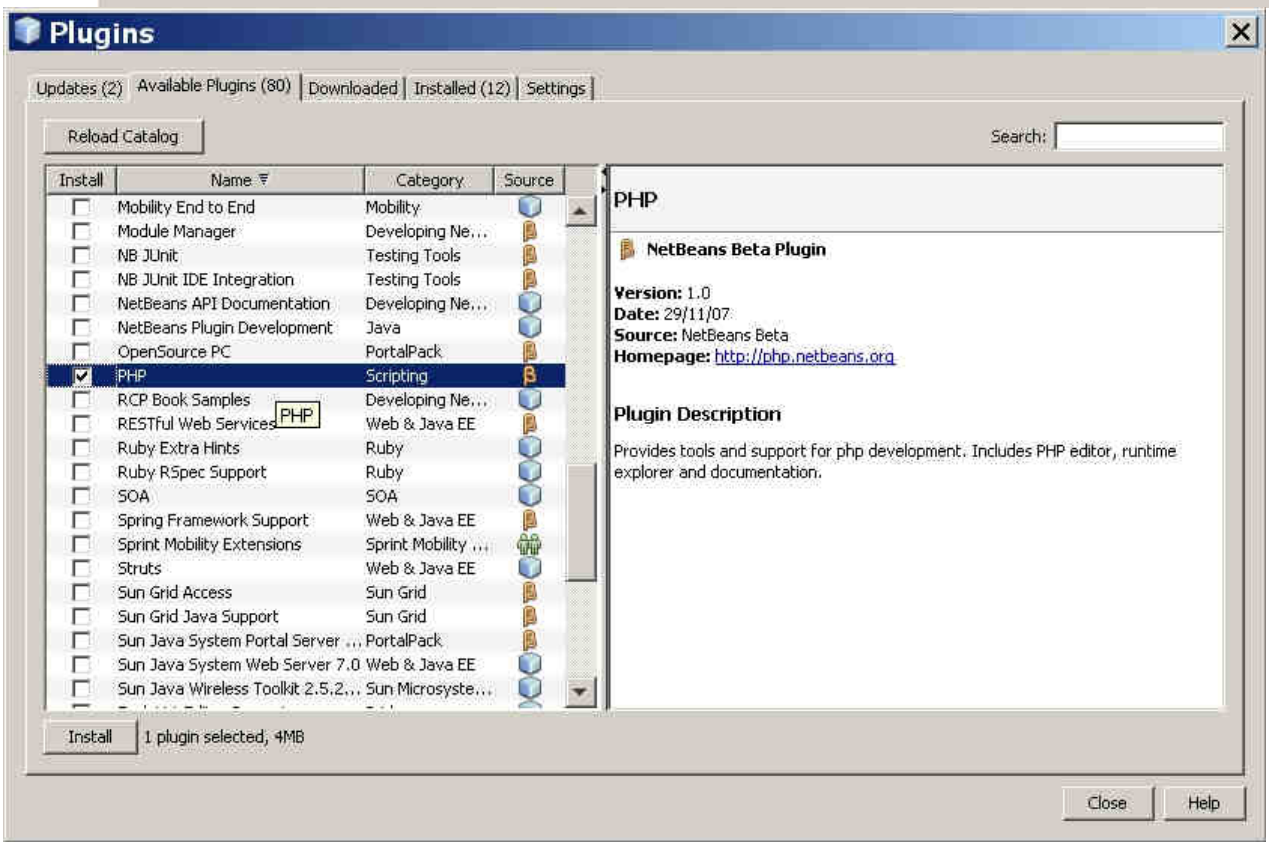


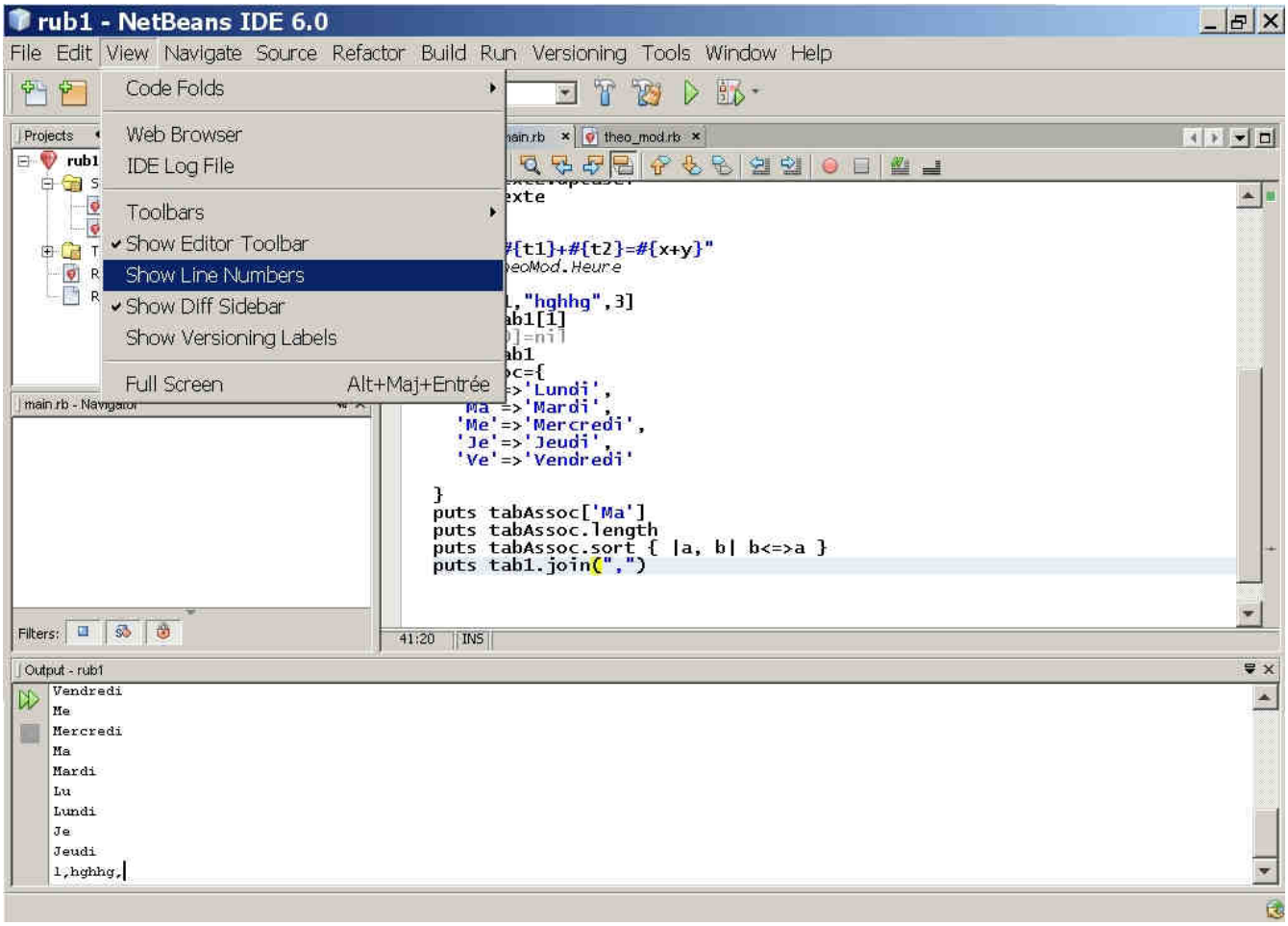



New Ruby Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project Name: rub1
Project Location: C:\applis
Browse...
Project Folder: C:\applis\rub1
Set as Main Project
Create Main File main.rb
Ruby Interpreter: Built-in JRuby (1.0.2)
Change...
< Back
Next >
Finish
Cancel
Help


rub1 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Versioning Tools Window Help
Projects
Files
Services
rub1
Source Files
main.rb
Test Files
Rakefile
README
Start Page
main.rb
To change this template, choose Tools | Templates
and open the template in the editor.
puts "Hello World"
main.rb - Navigator
Filters:
4:1
INS
Output - rub1
Hello World
main.rb saved.


rub1 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Versioning Tools Window Help
Palette
Add to Favorites
Apply Diff Patch...
Libraries
Ruby Gems
Templates
DTDs and XML Schemas
Plugins
Options
Projects
Files
Services
rub1
Source Files
main.rb
Test Files
Rakefile
README
Start Page
main.rb *
puts "Hello World"
main.rb - Navigator
Filters:
5:1
INS
Output - rub1
Hello World


Plugins
Updates (2)
Available Plugins (80)
Downloaded
Installed (12)
Settings
Reload Catalog
Search:
Install
Name
Category
Source
Mobility End to End
Module Manager
NB JUnit
NB JUnit IDE Integration
NetBeans API Documentation
NetBeans Plugin Development
OpenSource PC
PHP
RCP Book Samples
RESTful Web Services
Ruby Extra Hints
Ruby RSpec Support
SOA
Spring Framework Support
Sprint Mobility Extensions
Struts
Sun Grid Access
Sun Grid Java Support
Sun Java System Portal Server ...
Sun Java System Web Server 7.0
Sun Java Wireless Toolkit 2.5.2...
Mobility
Developing Ne...
Testing Tools
Testing Tools
Developing Ne...
Java
PortalPack
Scripting
Developing Ne...
Web & Java EE
Ruby
Ruby
SOA
Web & Java EE
Sprint Mobility ...
Web & Java EE
Sun Grid
Sun Grid
PortalPack
Web & Java EE
Sun Microsyste...
PHP
PHP
NetBeans Beta Plugin
Version: 1.0
Date: 29/11/07
Source: NetBeans Beta
Homepage: http://php.netbeans.org
Plugin Description
Provides tools and support for php development. Includes PHP editor, runtime explorer and documentation.
Install
1 plugin selected, 4MB
Close
Help


rub1 - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Versioning Tools Window Help
Code Folds
Web Browser
IDE Log File
Toolbars
Show Editor Toolbar
Show Line Numbers
Show Diff Sidebar
Show Versioning Labels
Full Screen Alt+Maj+Entrée
Projects
main.rb
theo_mod.rb
main.rb - Navigator
Filters:
41:20 INS
Output - rub1
Vendredi
Me
Mercredi
Ma
Mardi
Lu
Lundi
Je
Jeudi
1,hghhg,

IV. Le langage

A. Hello the World

```
#!c:/ruby/bin/ruby -w
puts "Hello, World!"
```

B. Paramètres de la ligne de Commande

```
puts ARGV.join('-')
```

C. Les chaînes

1. Here Doc

```
texte = <<EOF
Ceci est un test
EOF
```

2. Extraction

```
texte="Ceci est un texte"
puts texte.slice(5,3)
puts texte[5,3]
```

3. Quelques fonctions

```
"Test" + "Test   "TestTest
"Test".capitalize Test
"Test".downcase          test
"Test".chop      Tes
"Test".hash      -98625764
"Test".next      Tesu
"Test".reverse   tseT
"Test".sum       416
"Test".swapcase tEST
"Test".upcase    TEST
"Test".upcase.reverse     TSET
"Test".upcase.reverse.next          TSEU
```

4. Interpolation

```
puts "#{x} + #{y} = #{x + y}"
```

5. Expression régulière

```
^         Début de chaîne
$         Fin de chaîne
\A Anchor for thestart of a string
\Z Anchor for the end of a string
. Any character
\w        lettre, chiffre, tiret bas
\W        symétrique à \w
\d        N'importe quel chiffre
\D        Autre que des chiffres
\s        Tab, espace, fin de ligne
\S        Symétrique à \s

*         0, n occurrences
+         1, n occurrences
?         N'importe quel caractère
{x}       x occurrences
```

{x,y} entre x et y occurrences

D. Les fonctions

```
#!c:/ruby/bin/ruby -w
texte="Denis Szalkowski"
nombre=-1942
puts texte.length
puts texte.index("z")
puts nombre.abs
def aQueCoucou(monNom)
# result = "A que coucou,\n " + monNom
  result = "A que coucou,\n #{monNom}"
  return result
end
puts aQueCoucou(texte)
```

E. Variables

1. Nommage

Lettres, chiffres (sauf 1ère position), tiret bas

2. Préfixe

$ variables globaes
@@ instances
@ classe

F. Tableaux

```
#!c:/ruby/bin/ruby -w
arrayA = [ 1, 'un texte', 3.14 ]
puts arrayA[0]
arrayA[2] = nil
puts arrayA
arrayEmpty1 = []
arrayEmpty2 = Array.new
arrayB = %w{ Ceci est un test }
puts arrayB[0]
puts arrayB[3]
assoC = {
'0' => 'Dimanche',
'1' => 'Lundi'
}
puts assoC['0']
```

fonctions length et type, sort, reverse, join

G. Date

```
t=Time.now
puts t
puts t.sec, t.min, t.hour, t.day, t.month, t.year, t.wday, t.yday, t.isdst, t.zone
```

H. Structure du contrôle

1. Alternative

```
if...
elseif...
else
end
```

2. Répétitives

```
while...
end
until ...
end
```

3. Ecritures simplifiées

```
instruction if condition
instruction while condition
```

4. expression ternaire

```
age = 10
type = age < 18 ? "enfant" : "adulte"
puts "Vous êtes " + type
```

5. Case

```
fruit = "orange"
case fruit
when "orange"
color = "orange"
when "apple"
color = "green"
when "banana"
color = "yellow"
else
color = "unknown"
end
```

6. Gestion de blocs

```
#!c:/ruby/bin/ruby -w
def callBlock
  yield
  yield
end
callBlock { puts "Hello" }
```

7. Itération

```
#!c:/ruby/bin/ruby -w
jours = %w( Lun Mar Mer Jeu Ven Sam Dim )
jours.each { |jour| puts jour }
jours.each do |jour|
  print jour, " -- "
end
puts "\n"
5.times {  print "*" }
puts "\n"
0.upto(9) {|i|  print i }
puts "\n"
('a'..'z').each {|car| print car }

1.upto(5) { ...code to loop here... }
10.downto(5) { ...code to loop here... }
0.step(50, 5) { ...code to loop here... }
```

8. Gestion d'erreurs

```
begin
```

```
  f = File.open('c:\boot.ini')
            f.each { |l| puts "#{l}" }
rescue
  puts "fichier inexistant"
else
  puts "pas d\'erreur"
ensure
  f.close unless f.nil?
end
```

9. Les modules

a) Module

```
module Modinfo
  def info
    "info"
  end
end
```

b) Require et Load

Par défaut, la commande Require pointe vers C:\ruby\lib\ruby\1.8 ou le répertoire courant.
Vous devez employer ou load ou require.

```
load "modinfo.rb"
require "Modinfo"
class Test
  def initialize(prop)
    @prop=prop
  end
  include Modinfo
end
oTest=Test.new("test")
puts oTest.info
```

V. Les objets

A. Classe statique

```
class Fixnum
def seconds
self
end
def minutes
self * 60
end
def hours
self * 60 * 60
end
def days
self * 60 * 60 * 24
end
end
puts Time.now
puts Time.now + 10.minutes
puts Time.now + 16.hours
puts Time.now - 7.days
```

B. Création d'une classe

```
#!c:/ruby/bin/ruby -w
class Tiers
      attr_reader :rs, :cp, :ville
    def initialize(rs, cp, ville)
    @rs=rs
    @cp=cp
    @ville=ville
  end

end
```

C. La portée

Les méthodes ou propriétés peuvent être privées, publiques ou protégées.

```
#  public    :method1, :method4
 # protected :method2
 # private   :method3
```

D. Création d'un objet

```
oTiers = Tiers.new("Dsfc", 27800,"Saint Eloi de Fourques")
puts oTiers.inspect
puts oTiers.to_s
puts oTiers.rs
```

E. Héritage

```
class Client < Tiers
    attr_writer :credit
    attr_reader:credit

  def initialize(rs, cp, ville,credit)
    super(rs, cp, ville)

    @credit = credit
```

```
 end
 end
```

F. Utilisation d'une classe héritée

```
oClient = Client.new("Dsfc", 27800,"Saint Eloi de Fourques",34000)
oClient.credit=56000
puts oClient.credit
```

G. Lire une propriété

```
#!c:/ruby/bin/ruby -w
class Tiers

 def initialize(rs, cp, ville)
   @rs=rs
   @cp=cp
   @ville=ville
 end
 def get_rs
   @rs

   end
   def rs
return @rs
end
end
oTiers=Tiers.new("test",79000,"Niort")
puts oTiers.get_rs
puts oTiers.rs
```

H. Classe statique

```
class Stat
  class << self
    define_method :text1 do
      "test1"
    end
  end
  def self.text2
    "test2"
  end

 end
puts Stat::text2
```

I. connexion à MySql

1. La classe

```
load "C:/ruby/lib/ruby/gems/1.8/gems/activerecord-2.0.2/lib/active_record/vendor/mysql.rb"
#require "mysql"
class MonMysql
 @@res=nil

 def initialize(host,user,pwd,bd)

   begin
     @@conn = Mysql.real_connect(host,user,pwd,bd)
   rescue Mysql::Error => e
    puts "#{e.errno} : #{e.error}"
```

```
    end
  end
  def req(sql)
    @@res=@@conn.query(sql)
  end
  def liste
#   i=@res.num_rows
#   while row = @res.fetch_row do
#     printf "%s, %s\n", row[0], row[1]
#   end
   @@res.each do |row|
     printf "%s, %s\n", row[0], row[1]
   end
  end
  def destroy
    if @@res
      @@res.free
      @@res=nil
      puts "Libération mémoire"
    end
    puts "On ferme"
    @@conn.close
    @@conn=nil
  end
end
```

2. Le code

```
require "monmysql"
oMySql=MonMysql.new("localhost","root","root","crm_development")
oMySql.req("SELECT * from clients")
oMySql.liste
oMySql.destroy
oMySql=nil
```

VI. Rails

A. Installation de WebBricks

gem install rails--include-dependencies
gem update rails
rails -d mysql demo
cd demo
ruby script/server
ruby script/server --port=80

B. Structure de Rails

app Code

C. Création d'un contrôleur

ruby script/generate controller Say
Ruby crée le fichier app/controllers/say_controller.rb.

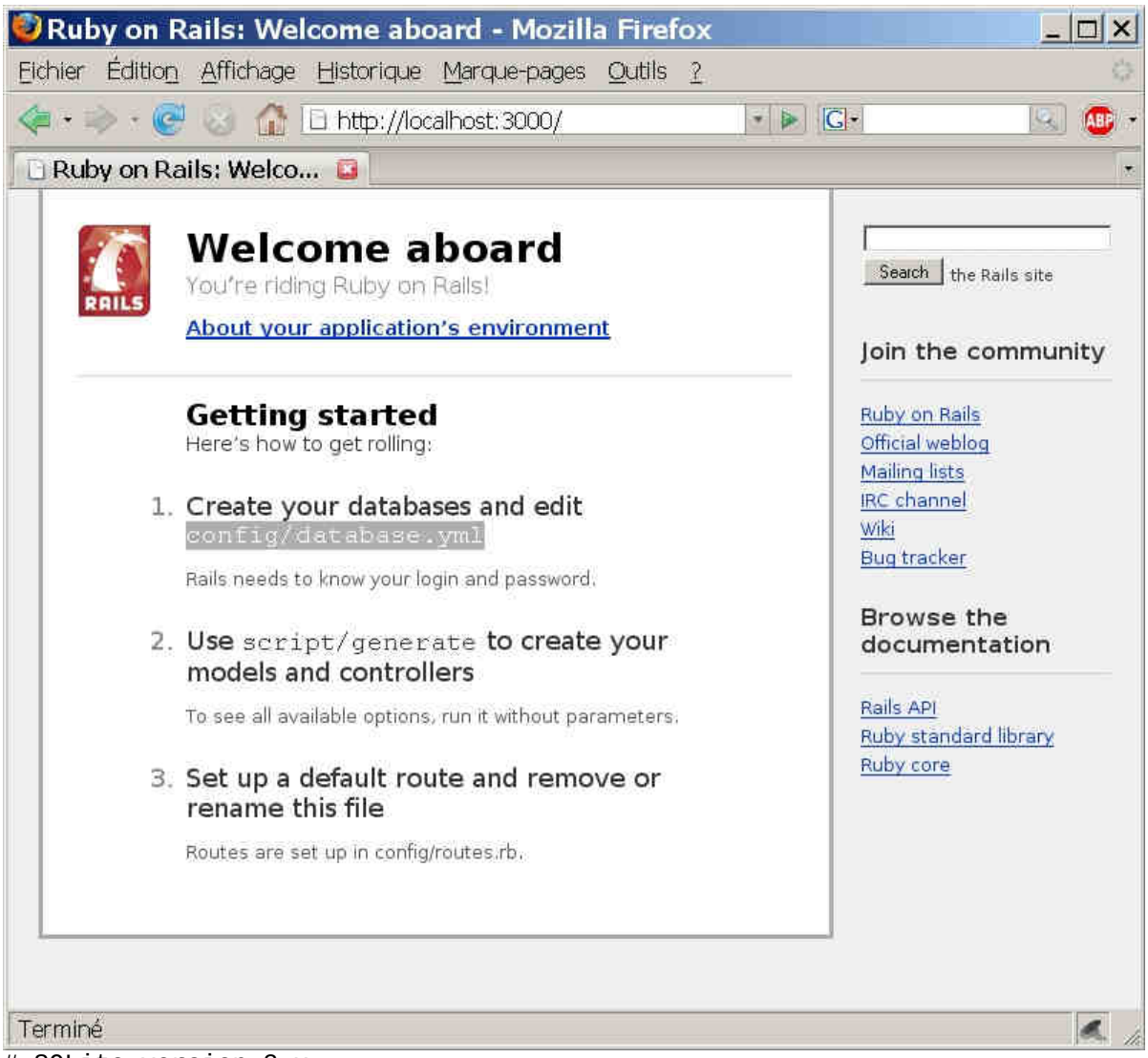
```
class SayController<ApplicationController
def hello
end
end
```

D. La vue

app/views/say/hello.rhtml

```
<html>
<head>
<title>Hello, Rails!</title>
</head>
<body>
<h1>Hello from Rails!</h1>
<p>
<%=@time%>.
</p>
<%=link_to "Hello from rails",:action=> "hello" %>
</body>
</html>
```

E. Configuration de la base




```
# SQLite version 3.x
#   gem install sqlite3-ruby (not necessary on OS X Leopard)
#development:
#  adapter: sqlite3
#  database: db/development.sqlite3
#  timeout: 5000

# Warning: The database defined as 'test' will be erased and
# re-generated from your development database when you run 'rake'.
# Do not set this db to the same as development or production.
#test:
#  adapter: sqlite3
#  database: db/test.sqlite3
#  timeout: 5000

#production:
#  adapter: sqlite3
#  database: db/production.sqlite3
#  timeout: 5000
login: &login
  adapter: mysql
  username: root
  password: root
  host: localhost

development:
```

```
  <<: *login
  database: development

test:
  <<: *login
  database: test

production:
  <<: *login
  database: production

development:
  adapter: mysql
  database: crm_development
  user: root
  password: root
test:
  adapter: mysql
  database: crm_test
  user: root
  password: root

production:
  adapter: mysql
  database: crm_production
  user: root
  password: root
```

F. création des databases et des tables

1. Création des bases

rake db:create:all

2. Création du script de crétaion de tables

ruby script/generate scaffold Client rs:string ville:string

Rails génère une table à l'aide du script db/migrate/001_create_clients.rb

```
class CreateClients < ActiveRecord::Migration
 def self.up
  create_table :clients do |t|
   t.string :rs
   t.string :ville

   t.timestamps
  end
 end

 def self.down
  drop_table :clients
 end
end
```

3. Modification de table

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.column :login, :string
      t.column :password, :string
      t.column :email, :string
    end
   add_column :classifieds, :user_id, :integer
  end

  def self.down
    drop_table :users
    remove_column :classifieds, :user_id
  end
end
```

4. Type de champ

string
integer
text
float
datetime
date
time
timestamps
binary

5. Création des tables

rake db:migrate

G. Création d'un contrôleur

ruby script/generate controller Say
Ruby crée le fichier app/controllers/say_controller.rb.
class SayController<ApplicationController
def hello
end
end

H. La vue

app/models/client.rb
class Client < ActiveRecord::Base
validates_presence_of :rs, :ville
validates_presence_of : cp, :message=>"Code postal obligatoire"
validates_length_of :rs, :within=>1..20
validates_length_of :rs, :maximum=>30
validates_uniqueness_of :rs, :message => "La ville existe"
validates_numericality_of :cp
validation_confirmation_of :password , :message=>"Retapez le mot de passe"
validates_acceptance_of :check, :message=>"A cocher"
protected
def validate
errors.add(:cp, "Prix obligatoire") if cp >=1000 || cp <97999

```
end
end
```

I. Les lookups

```
<p><label for="classified_category">Category</label><br />
<%= collection_select(:classified, :category_id, @categories,
:id, :name) %></p>
```

J. La gestion des formulaires

1. La page text.html

Elle doit être stockée dans le répertoire public de l'application Rails

```
<html>
<head>
<title>Page Text</title>
</head>
<body>

<form action = "/text/nom" >
<!-- <form action = "/text/nom" method = "post" >-->
Votre nom<br />
<input type="text" name="txtNom">
<input type="submit" />
</form>
</body>
</html>
```

2. Le contrôleur

ruby script/generate controler Text

Editez le fichier /app/controllers/text_controller.rb

```
class TextController < ApplicationController
  def nom
    @data = params[:txtNom]
  end
end
```

3. La vue

Le code doit être sauvegardé dans le fichier /app/views/text/**nom**.rhtml

```
<html>
<head>
<title>Lecture du contenu du champ</title>
</head>
<body>
Vous êtes<%= @data %>.
</body>
</html>
```

4. Le formulaire Ruby

```
<html>
<head>
</head>
<body>
<%= start_form_tag ({:action => nom}, {:method => post}) %>
Entrez votre nom
<br />
<%= text_field_tag (txtNom, , {size => 30}) %>
```

<input type="submit" />
<%= end_form_tag %>
</body>
</html>

K. La feuille de style

/public/stylesheets/style.css

L. Conversion

<%=number_to_currency(@champ) %>
number_to_human_size
number_to_percentage
number_to_phone
champ.strftime("%d %B %Y")

M. Envoi de mail

Editez le fichier environnement.rb

```
ActionMailer::Base.delivery_method = :smtp

ActionMailer::Base.server_settings = {
:address => "smtp.railssolutions.com",
:port => 25,
:domain => "railssolutions.com",
:authentication => :login,
:user_name => "username",
:password => "password",
}
```

VII. Ressources

http://fairleads.blogspot.com/2007/12/rails-20-and-scaffolding-step-by-step.html
http://florent.sabourin.eu/rails/rails.html
PAge 59 Agile Web Development With Ruby On Rails.pdf