

Rational Build Forge



Tutorial 6: Dot commands in Rational Build Forge

Version 7.1.3

Note

Before using this information and the product it supports, read the information in "Notices," on page 9.

This edition applies to version 7.1.3 of Rational Build Forge and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2011.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Tutorial 6: Dot commands in Rational Build Forge

With Dot commands, you can access the capabilities and functions of the system. You can use multiple dot commands in an IBM® Rational® Build Forge® step and intersperse them with ordinary commands.

The tutorial shows you how to use the following commands:

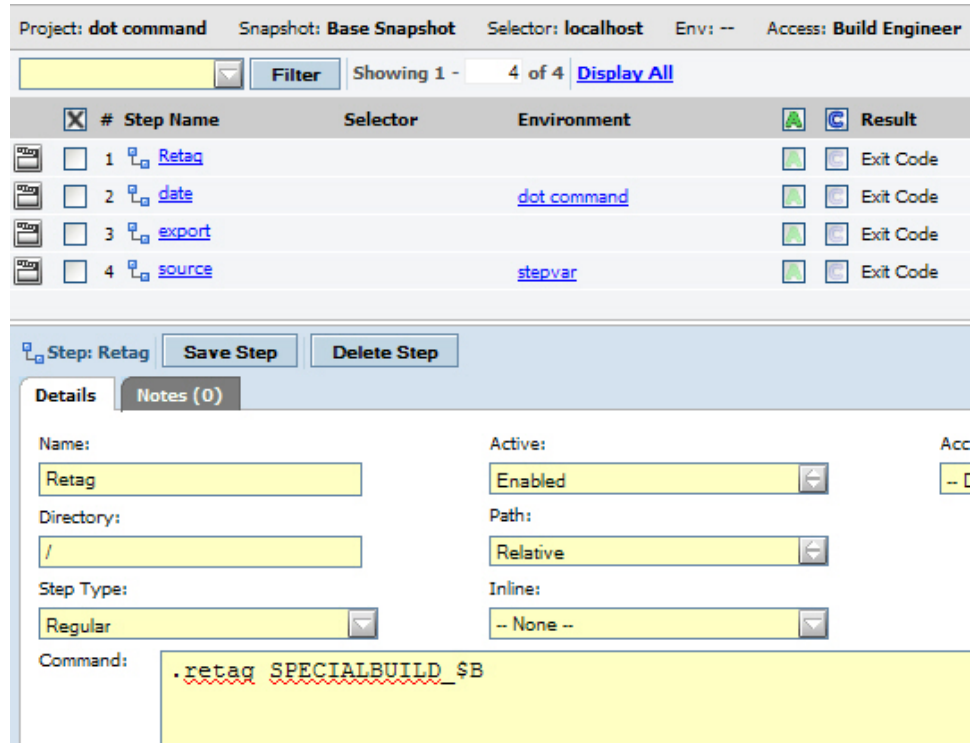
- “Changing build tags with the `.retag` command”
- “Accessing system information with the `.date` command” on page 2
- “Moving files with the `.put` and `.get` commands” on page 4
- “Changing a selector with the `.bset` command” on page 5
- “Exporting a project with the `.export` command” on page 5
- “Using a `.source` command in an environment variable” on page 6

Time required: 45 minutes

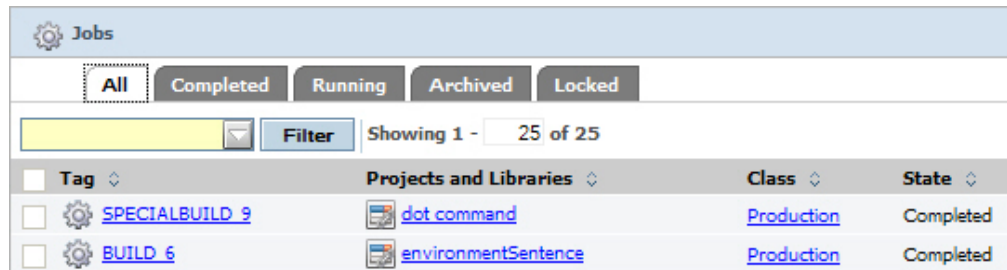
Changing build tags with the `.retag` command

The default build tag that is displayed in the Jobs list is `Build_N` where `N` is the current build number. The `.retag` command changes the build tag for the given project. With this command, you can specify another name, such as `SPECIALBUILD_1`, `SPECIALBUILD_2`, and so on.

1. Create a project, named dot commands, with the appropriate selector and access groups. You learned how to create projects and steps in Tutorial 1: Creating, running, and scheduling a Rational Build Forge project. For information about access groups, see Tutorial 3: Rational Build Forge administration: Controlling user access.
2. Create a step named Retag.
3. On the Details page, in the **Command** field, type the following code:
`.retag SPECIALBUILD_$$B`



4. Click **Save Step**.
5. Run the *dot commands* project and view the results.

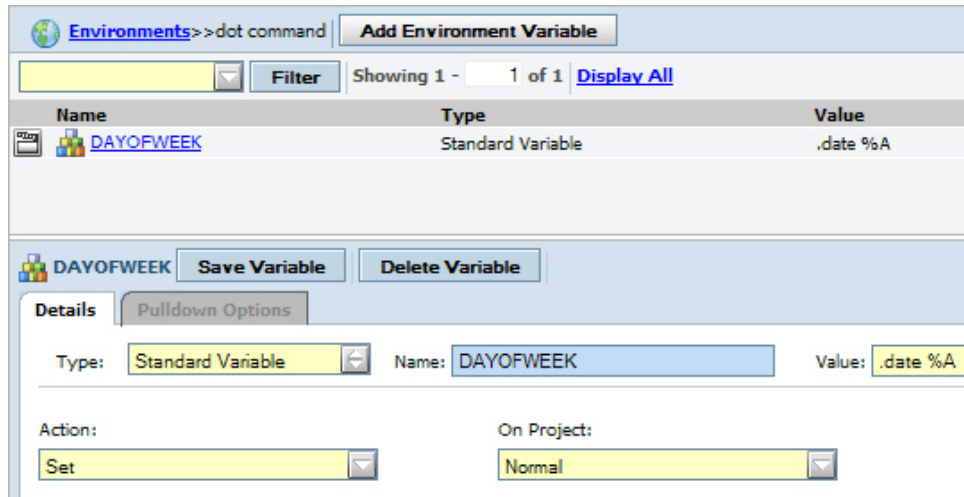


Accessing system information with the .date command

With the `.date` command, you can access system information about the current day of the week, date, and time.

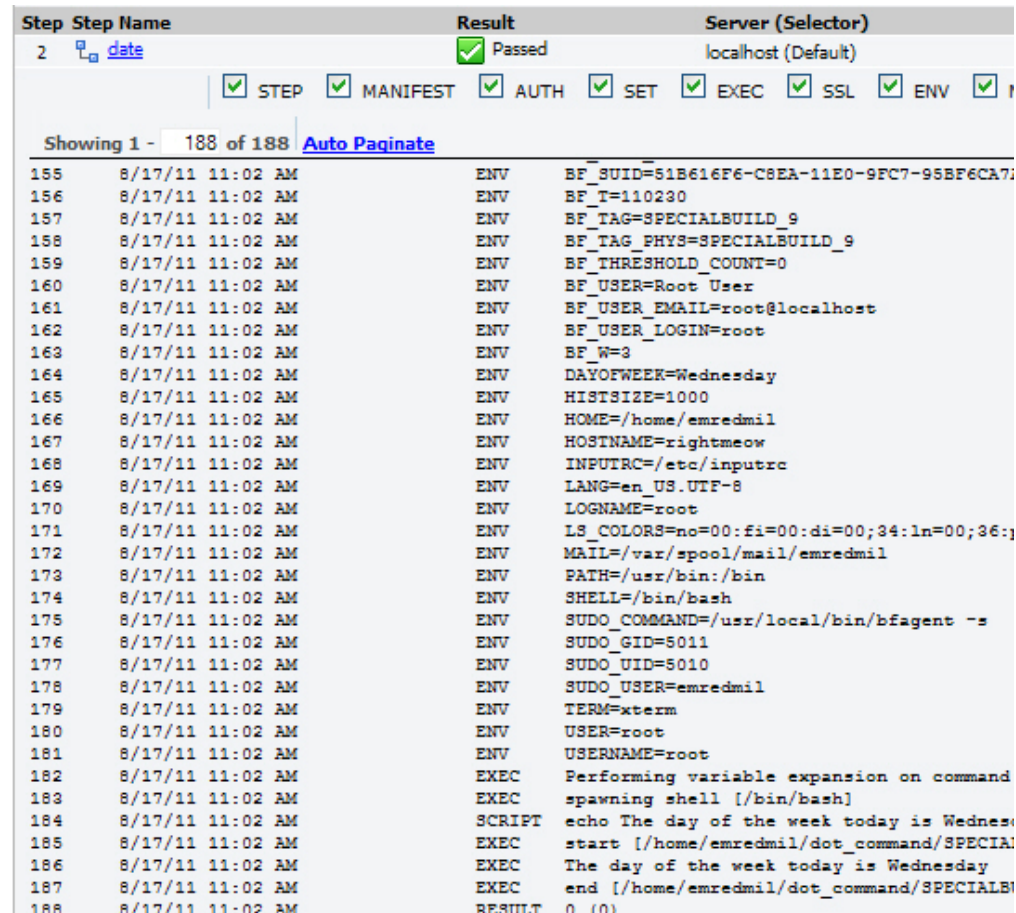
1. Create an environment named `dot command`.
2. Create a new standard environment variable, named `DAYOFWEEK`, with a value of `.date %A`. This value returns the full name of the current day of the week.

Tip: For information on other `.date` commands, see the topics in the Rational Build Forge information center.



3. In the *dot commands* project, create a new step named `date`.
4. On the Details page, in the **Command** field, type the following code:
`echo The day of the week today is ${DAYOFWEEK}`
5. Set **Environment** to **dot command**.
6. Run the *dot commands* project.

The following image shows the output from the *dot commands* project.



Moving files with the .put and .get commands

The `.put` and `.get` commands are used to send and retrieve files between two servers.

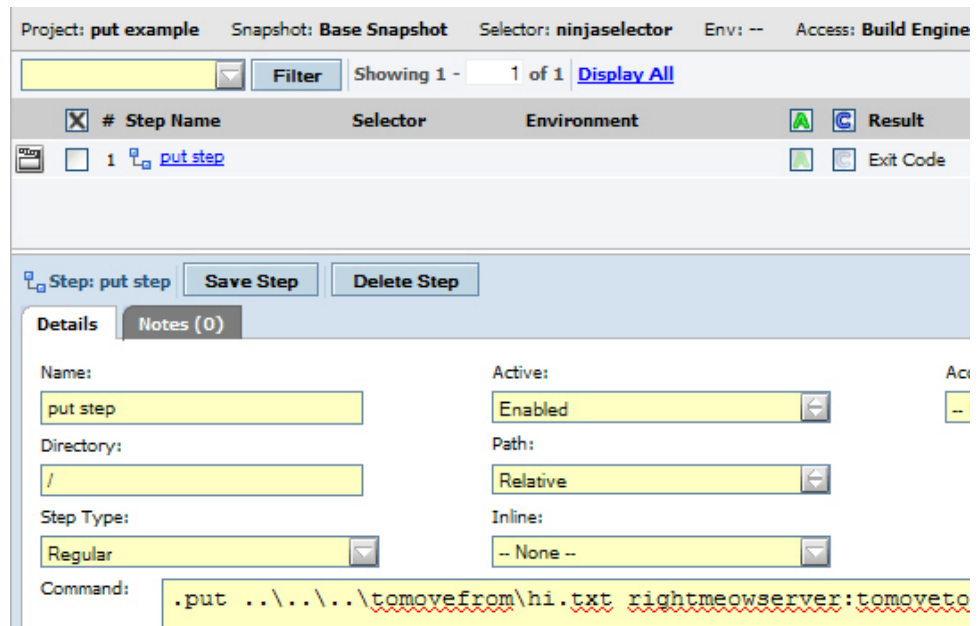
To complete this lesson, you need access to two working servers. The example uses *ninjaserver* as the server containing the file to be moved and *rightmeowserver* as the server that is the target of the moved file.

Restriction: Do not use these commands to transfer large files.

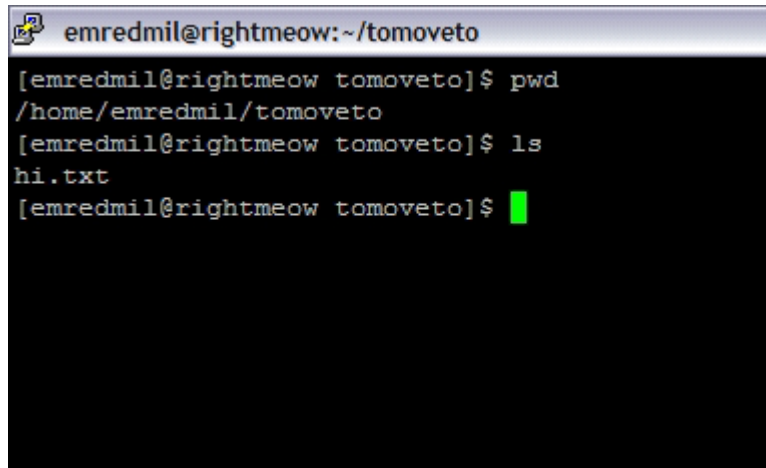
1. On *ninjaserver*, create a blank file called `hi.txt`.
2. In Rational Build Forge, create a project named `put`.
3. Create a step, named `putstep`, and in the **Command** field type the following code:

```
.put ../../..\tomovefrom\hi.txt rightmeowserver:tomoveto/hi.txt
```

Note: The path is specific to the target location of the file to be moved. A `put` command uses the following syntax: `.put [<relative_path>/]file server:[[<relative_path>/]file]`



4. Run the project.
5. Navigate to the `tomoveto` directory on the *rightmeow* server and see that the `hi.txt` file has been moved there. The following image shows an example.



```
emredmil@rightmeow:~/tomoveto
[emredmil@rightmeow tomoveto]$ pwd
/home/emredmil/tomoveto
[emredmil@rightmeow tomoveto]$ ls
hi.txt
[emredmil@rightmeow tomoveto]$
```

Changing a selector with the `.bset` command

The `.bset` command is used to change the value of an environment variable or to change the server, selector, or buildserver used by the project. All steps after the step implementing `.bset` are affected by the changes made in that step.

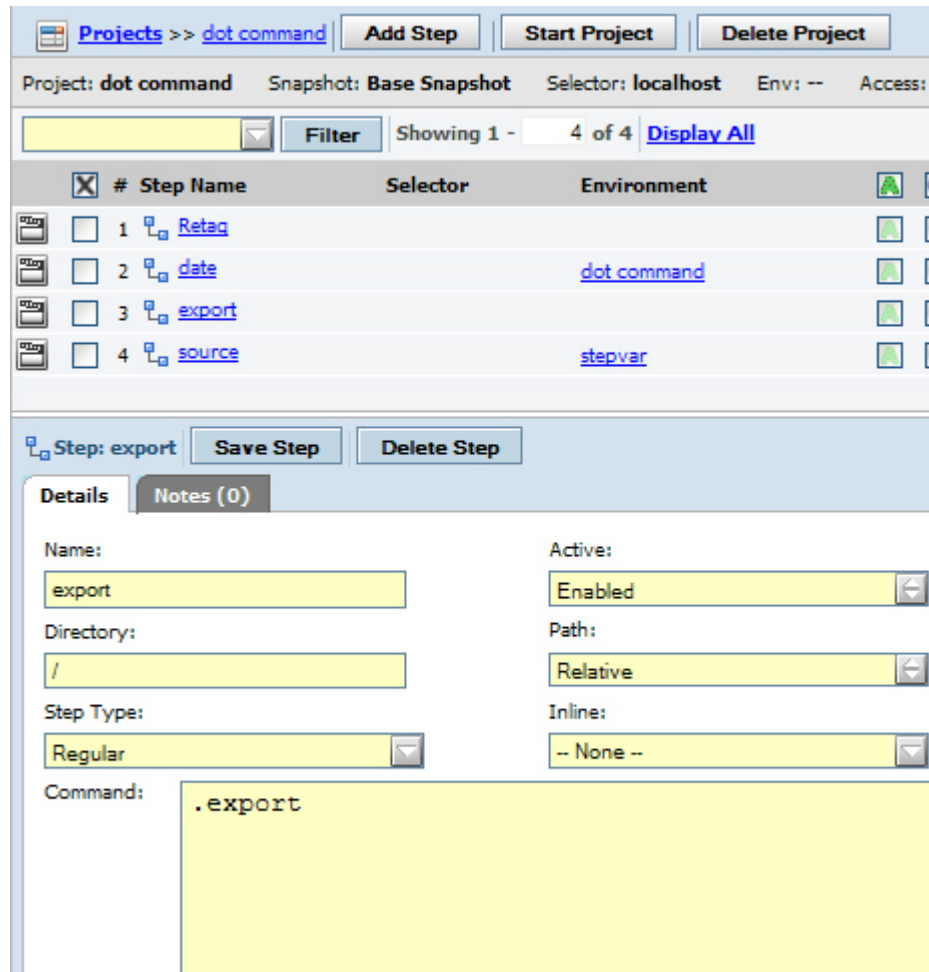
1. In the *put* project, create a step named *bsetstep*:
 - a. Click **Add Step**.
 - b. In **Name**, type *bsetstep*.
 - c. In **Command**, type the following code:

```
.bset rightmeowselector
```
 - d. Click **Save Step**.
2. Create another new step named *echostep*, in **Command** type `echo hi`, and then click **Save Step**. This Rational Build Forge step will show that the selector was changed.
3. Run the project and view the results.

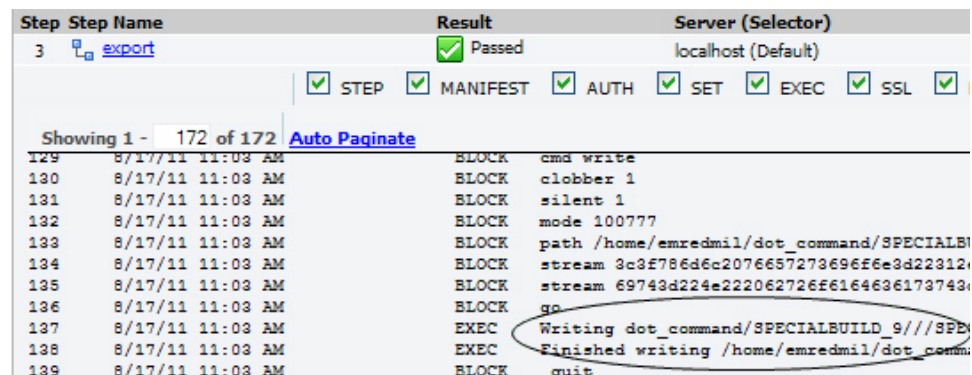
Exporting a project with the `.export` command

The `.export` command saves the project definition for the calling project to an XML file located in the working directory of the step.

1. In the *dot commands* project, create a step named *export*, in **Command** type `.export`, and then click **Save Step**.



2. Run the project. In the log, the location of the file is listed, as shown in the following image.



Using a .source command in an environment variable

You can use the **.source** command with both adapters and environments. We will address using the command with batch files. This lesson shows how to use the **.source** command in an environment variable for both Windows and Linux systems.

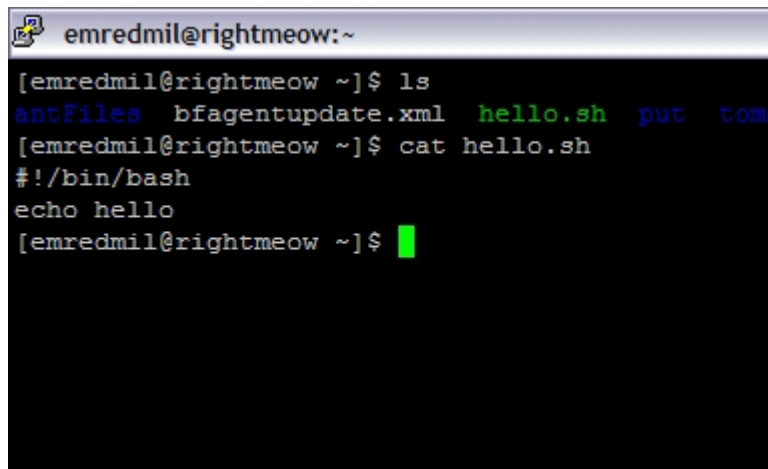
To implement **.source** in an environment variable on Windows systems:

1. Create a file, hi.bat, in the C:/temp directory. The file contains the command echo hello.
2. Create an environment named stepvars.
3. Create a standard environment variable:
 - a. In **Name**, type .source.
 - b. In **Value**, type C:/temp/hi.bat.
 - c. Click **Save Variable**.
4. In the dot commands project, create a new step named sourcestep. Then, in **Command** type echo goodbye and in **Environment** select stepvars.
5. Run the project and view the result.

Note: The batch script is executed before the step command.

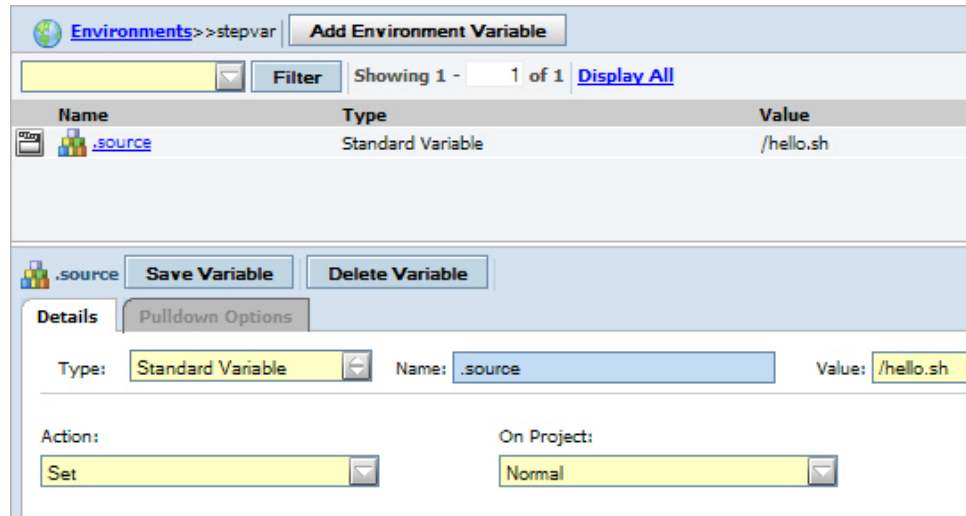
To implement .source in an environment variable on Linux systems:

1. In your home directory, create a file named "hello.sh" with the commands shown below.



```
emredmil@rightmeow:~  
[emredmil@rightmeow ~]$ ls  
antFiles  bfaagentupdate.xml  hello.sh  put  tom  
[emredmil@rightmeow ~]$ cat hello.sh  
#!/bin/bash  
echo hello  
[emredmil@rightmeow ~]$
```

2. Create new environment named stepvars.
3. Create a standard environment variable:
 - a. Click **Add Environment Variable**.
 - b. In **Name**, type .source.
 - c. In **Value**, type /hello.sh.
 - d. Click **Save Variable**.



4. In the *dot commands* project, create a new step named `sourcestep`.
 - a. Click **Add Step**.
 - b. In **Name**, type `sourcestep`.
 - c. In **Command**, type the following code: `echo goodbye`.
 - d. In the **Environment** list, select **stepvars**.
 - e. Run the project and view the results.

Note: The bash script is run before the step command.

Tutorial summary

Now that you have completed this tutorial, you can begin implementing Dot commands for your Rational Build Forge projects.

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.html.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.