

Version 7.0.1

Windows, UNIX, and Linux

IBM Confidential



Storing History Dates in Coordinated Universal Time

Before using this information, be sure to read the general information under "Notices" on page 13.

June 2007

This edition applies to IBM Rational ClearQuest Version 7.0.1 and ClearQuest MultiSite, and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1992, 2006. All rights reserved.**

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

Contents

Storing history dates in Universal Coordinated Time	1
Background.	1
Terminology	1
Choosing an implementation strategy	2
Cross-version compatibility issues	2
Enabling the UTC feature and converting databases	3
Converting databases at one time	3
Converting databases in phases	3
Reference	5
timechange.pl	5
Configuration file.	7
Database changes after converting dates to UTC	8
installutil subcommands	9
getutchistorytimestamps	9
setutchistorytimestamps	10
ResultSet object methods	11
GetConvertToLocalTime	11
SetConvertToLocalTime	12
Notices	13

Storing history dates in Universal Coordinated Time

This paper describes the planning and implementation details for enabling storage of history dates in Rational ClearQuest records in a time-zone-independent format.

The audience for this paper is Rational ClearQuest administrators.

This paper supplements the information in the *IBM Rational ClearQuest and ClearQuest MultiSite Installation and Upgrade Guide*, the Rational ClearQuest administrator's help, the Rational ClearQuest command-line interface help, and the Rational ClearQuest API help.

Background

Rational ClearQuest software stores history dates for records in the history table of the user database. Changes to objects are recorded by inserting a row in the history table. The history table contains two date/time columns: `action_timestamp` and `expired_timestamp`. These time stamps have always been recorded in the local time of the database server, which is sufficient in a single-site environment. However, in a replicated deployment, where database servers are in different time zones, storing time stamps in the local time of the database server can cause actions to appear to have occurred out of order. For example, a defect might appear to be closed before it is opened or assigned before it is submitted.

The solution to this problem is to store Rational ClearQuest history time stamps in Coordinated Universal Time (UTC).

Terminology

This section defines important terms used in this paper:

local time

The time in the region where a machine is located, usually described as an offset from UTC.

GMT Greenwich Mean Time. A historical term, referring to mean solar time at the Royal Greenwich Observatory in Greenwich in London, England.

In many contexts in the Rational ClearQuest documentation, the terms GMT and UTC are used interchangeably.

UTC Coordinated Universal Time (UTC). A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal civil time all over the earth. Time zones around the world are expressed as positive and negative offsets from UTC.

In many contexts in the Rational ClearQuest documentation, the terms GMT and UTC are used interchangeably.

YYYY-MM-DD HH:MM:SS

ISO 8601 date-and-time notation used to store history time stamps in a Rational ClearQuest database.

Choosing an implementation strategy

New user databases created with Rational ClearQuest version 7.0.1 at feature level 6 automatically use UTC to store history dates.

However, the default for user databases created with Rational ClearQuest versions 2003.06.xx and 7.x at feature level 5 is to create time stamps using the local time of the database server. If you have an existing Rational ClearQuest environment with user databases created at feature level 5, you must decide whether to convert the history time stamps. For information about feature levels, see the *IBM Rational ClearQuest Installation and Upgrade Guide*, version 7.0.1.

Enabling your databases to store history dates in UTC requires planning. It is important to understand the impact of this feature before enabling it. You must decide whether to convert history time stamps in your existing databases to UTC, or to enable the UTC feature only for new records. Mixing old and new history time stamp formats will be noticeable in query, report, and chart results. However, it may not be important that current data is off by a few hours or occasionally out of sequence.

Cross-version compatibility issues

If you have user databases created with Rational ClearQuest version 7.0.1 at feature level 6, then you will already have upgraded all of your Rational ClearQuest clients to version 7.0.1. These clients will store and display records using the UTC time stamps.

However, if you have a Rational ClearQuest MultiSite environment running versions 2003.06.xx or 7.x at feature level 5, these clients cannot display UTC dates in local time. These clients continue to display dates as read from the database, meaning that dates are displayed differently in old and new clients. You will have more consistent and reliable results if you upgrade all clients to version 7.0.1 or later.

In a Rational ClearQuest MultiSite environment, where server systems are running Rational ClearQuest version 7.0.1, you must also consider the following compatibility issues before enabling the UTC feature:

- Supported clients that run an earlier version of Rational ClearQuest software can modify UTC-converted databases. These clients continue to store history dates in the database by using server local time, which can result in a mix of old and new date formats. You can correct these databases by running the UTC conversion script, **timechange.pl**. However, to ensure that new time data is stored in databases consistently, upgrade all your clients at the same time.
- Supported clients that run an earlier version of Rational ClearQuest software cannot display UTC dates in local time. These clients continue to display dates as read from the database, meaning that dates are displayed differently in old and new clients.
- After a site is converted, it can import history records from an unconverted site. Therefore, it is a good practice to synchronize all replicas before converting the databases and to coordinate the conversion to avoid running **timechange.pl** multiple times.

Yet, even if all replicas are synchronized, the UTC feature is enabled, and the databases are converted to UTC, a Rational ClearQuest MultiSite **restore replica**

command might cause old operation logs (oplogs) to be exported again. If so, after the oplogs are imported, you must run **timechange.pl** again to convert dates that are not in UTC.

Enabling the UTC feature and converting databases

If you have an existing Rational ClearQuest MultiSite environment, you must decide whether to convert all the databases to UTC at one time, or to convert them in phases.

Converting databases at one time

To enable the UTC feature for all clients and servers at one time, follow this sequence:

1. Follow the instructions in the *IBM Rational ClearQuest and ClearQuest MultiSite Installation and Upgrade Guide* to perform the following tasks:
 - Upgrade all clients and servers to version 7.0.1 of Rational ClearQuest
 - Disable user logons
 - Upgrade synchronization servers at all sites to Rational ClearQuest MultiSite version 7.0
 - Synchronize all sites
 - Back up your Rational ClearQuest schema repositories and vendor bases
2. At each server site:
 - a. Enable date storage in UTC by running the **installutil setutchistorytimestamps** subcommand. For details, see “installutil subcommands” on page 9.
 - b. Convert old dates to UTC using the **timechange.pl** script. For details, see “timechange.pl” on page 5.
 - c. Re-enable user logons.

After schema repositories and user databases are upgraded, users who have not upgraded their client machines can easily access Rational ClearQuest version 7.0.1 databases through the Rational ClearQuest Web client.

Converting databases in phases

To enable the UTC feature in phases, you must run the **timechange.pl** script multiple times. Enabling the UTC feature and converting your databases and in phases allows some sites to remain operational during the conversion and feature-enablement process. A disadvantage is that dates that have not been converted yet may be synchronized to sites that have already been converted, so you must run **timechange.pl** multiple times.

You must also restart the Rational ClearQuest clients after enabling UTC.

Reference

This section describes the following new script, subcommands, and methods that support storing history dates in user databases in UTC:

- “timechange.pl” script
- “installutil subcommands” on page 9:
 - **getutchistorytimestamps**
 - **setutchistorytimestamps**
- “ResultSet object methods” on page 11:
 - **GetConvertToLocalTime**
 - **SetConvertToLocalTime**

timechange.pl

Perl script that converts old history dates in a Rational ClearQuest database to UTC

Synopsis

```
cqperl timechange.pl
  -cmd readdb | checkdb | gensql | runsql | fixdb | verifycfg
  [ -help ] [ -verbose n ]
  [ -historyfile filename ] [ -sqlfile filename ] [ -idblockfile filename ]
  [ -pdsqlargs arguments ] [ -cfg config-filename ] [ -removeutc ]
```

Description

The **timechange.pl** script converts history dates in a Rational ClearQuest database to UTC and records the conversion in the comments field of the associated history table rows. The script can be run multiple times on the same database. It skips rows that it has updated. In a Rational ClearQuest MultiSite environment, **timechange.pl** is run at each site.

The **timechange.pl** script has three phases of operation:

- Reading information from the user database
- Analyzing the database information and generating repair commands
- Running the repair commands

When you check whether any history records require updating or generate repair commands, input to **timechange.pl** is a configuration file containing rows that describe date transformations to perform. The same configuration file can be used at more than one site. See “Configuration file” on page 7 for details.

Restrictions

You must have Rational ClearQuest administrator privileges to run this script.

Options and arguments

Options and associated arguments to **timechange.pl** are required for some commands but optional for others. For example, you do not need the **-cfg** option, specifying the input configuration file, for the **readdb** or **runsql** commands, but it

is required for the **checkdb**, **gensql**, **fixdb**, and **verifycfg** commands. Similarly, if you are analyzing previously read data (using the **checkdb** command) or generating repair commands (using the **gensql** command) but are not running repair commands, you do not need to specify the database connection arguments.

-cmd *command*

Command to run using the **timechange.pl** script. To perform more than one command, use a comma-separated list. For example:

```
cqperl timechange.pl -cmd readdb,checkdb
```

Following are valid values:

readdb

Reads the history and ratl_id_blocks tables from the database and writes the results to a file.

checkdb

Reads the file produced by the **readdb** command and reports whether the database needs to be repaired. Use with the **-cfg** *cfg-filename* option and argument.

gensql

Reads the file produced by the **readdb** command, reports whether the database needs to be repaired, and generates repair commands, if needed. Use with the **-cfg** *cfg-filename* option and argument.

runsql Runs the previously generated repair commands.

fixdb Runs the following script:

```
cqperl timechange.pl -cmd readdb,gensql,runsql
```

Use with the **-cfg** *cfg-filename* option and argument.

verifycfg

Validates the configuration file.

-help Prints the script usage message.

-verbose *n*

Sets the verbosity level. The higher the number, the more messages.

Valid values are 0–3; the default is 1.

Note: Because of the output volume, consider a setting of 2 or 3 only if you have a very small database.

-historyfile *hist-filename*

File that history records are written to and read from.

-sqlfile *sql-filename*

File that repair commands are written to and read from.

-idblockfile *idblock-filename*

File that ratl_id_blocks records are written to and read from.

-pdsqlargs *arguments*

Command-line arguments to use with the **pdsql** utility. The arguments, which must be quoted appropriately for the command-line processor (**cmd**, **tcsh**, **bash**, and so on), are different for each vendor database. For more information about **pdsql**, see the *IBM Rational ClearQuest and ClearQuest MultiSite Installation and Upgrade Guide*, and Technote 1145079 from the IBM

Software Support Home page for Rational products at
<http://www.ibm.com/software/rational/support/>.

-cfg *cfg-filename*

Name of the configuration file that contains information for mapping sites to time zones. For details, see "Configuration file."

-removeutc

Reverts dates stored in UTC to the local time of the database server.

Configuration file

The input configuration file for **timechange.pl** contains rows that describe date transformations to perform. The file specifies how much to adjust each date based on the site and the date value. You can adjust dates in aggregate, by adjusting all dates for a site by a specified number of hours, or with a finer level of detail, by adjusting date ranges for a site by a specified number of hours.

There are several reasons why the configuration file is needed as input to **timechange.pl**:

- In a single site, the date transformations must account for the daylight savings time rules for the database server. Different locations in a single time zone can have different rules. For example, in the United States, Indiana is in the Eastern Standard Time (EST) zone but does not use daylight savings time. Some parts of Arizona, for example, on the Navaho reservation, use daylight savings time, and other parts do not. Furthermore, each year has different start and end dates for daylight savings time.
- If a database server has changed time zones, different adjustments must be applied for dates during various periods.
- In a Rational ClearQuest MultiSite environment, you must determine at which site each database record ID was allocated and then apply different adjustments for different rows in the history table. Complications mentioned in the previous two paragraphs might also apply to each site in a Rational ClearQuest MultiSite environment. For example, one database server might have changed time zones at some point during the lifetime of the clan.

Figure 1 shows a sample configuration file. The dates for Site1 and Site2 are adjusted in aggregate, by subtracting 5 hours and adding 6-½ hours, respectively. Dates for Site3 that fall within the specified date/time ranges are adjusted by subtracting 5 or 4 hours, according to the corresponding rows in the configuration file. The dates for Site5, which are already in UTC, are marked accordingly.

```
Site: Site1 -5
Site: Site2 +6.5
Site: Site3 -5 2004-10-25 - 2005-03-29 02:00:00
Site: Site3 -4 2005-03-29 02:00:00 - 2005-10-31 02:00:00
Site: Site3 -5 2005-10-31 02:00:00 - 2006-04-01 02:00:00
Site: Site5 0
```

Figure 1. Sample configuration file for timechange.pl

Note: If the Rational ClearQuest Export and Import Tools were used to put history records in the database, the time zones in which the records originated or were modified cannot be determined because the record IDs of the imported history records are from the importing site. The original record IDs are from a different series of IDs at the exporting site. If it is important to preserve

the history dates of imported records, the exporting database must be converted before the records are exported.

Database changes after converting dates to UTC

When the UTC feature is enabled and history dates in user databases are converted to UTC, information is included in the comments field of the associated history table row. The time zone of the database servers that store the dates is also recorded.

Table 2 gives examples of how the comments field of a history table row is updated. Table 1 describes the shorthand notation used in Table 2.

Table 1. Shorthand notation used in Table 2

Shorthand notation	Meaning
A	action_timestamp
E	expired_timestamp
P	Programmatically generated dates, such as those generated by Rational ClearQuest software. In the examples in Table 2, all time stamps were generated programmatically.
C	Dates converted to UTC by timechange.pl .
<i>version number</i>	A version number is added to allow for future revisions in this scheme. In the examples in Table 2, all time stamps are at version 1.

Table 2. Comments field examples after converting to UTC

Comments field	Explanation
AP1 -05:00, EP1 06:30	The action_timestamp is stored in UTC and the database server that stored the date was 5 hours west of Greenwich. The expired_timestamp also is in UTC and the database server that stored it was 6-1/2 hours east of Greenwich.
AP1 -05:00	The action_timestamp is stored in UTC and the database server that stored the date was 5 hours west of Greenwich. The expired_timestamp either is NULL or is stored in database server local time.
, EP1 03:00	The action_timestamp is stored in database server local time. Note: action_timestamp values are never NULL. The expired_timestamp is stored in UTC and the database server that stored it was 3 hours east of Greenwich.
NULL	Neither the action_timestamp nor the expired_timestamp is in UTC. The expired_timestamp may be NULL.

Examples

Following are several examples of running **timechange.pl** in a DOS session with different options and arguments. The script, options, and arguments must be entered on one line.

- Read the history and ID block information from the database.

```
cqperl timechange.pl -cmd readdb -hist \temp\history.out -idbl
\temp\idblocks.out -pdsqlargs "-v access -u admin -db \cqtest\sample.mdb"
```
- Generate repair commands, if needed.

```
cqperl timechange.pl -cmd gensql -hist \temp\history.out -idbl
\temp\idblocks.out -sqlfile \temp\fixdb.sql -cfg \temp\cfg.in
```
- Run the repair commands generated in the previous example.

```
cqperl timechange.pl -cmd runsql -sqlfile \temp\fixdb.sql -pdsqlargs "-v
access -u admin -db \cqtest\sample.mdb"
```
- Perform the previous three commands by combining them in one command.

```
cqperl timechange.pl -cmd fixdb -hist \temp\history.out -idbl
\temp\idblocks.out -sqlfile \temp\fixdb.sql -cfg \temp\cfg.in -pdsqlargs
"-v access -u admin -db \cqtest\sample.mdb"
```
- Check whether any history records need to be updated.

```
cqperl timechange.pl -cmd readdb,checkdb -hist \temp\history.out -idbl
\temp\idblocks.out -cfg \temp\cfg.in -pdsqlargs "-v access -u admin -db
\cqtest\sample.mdb"
```

installutil subcommands

The **installutil** command line utility sets up and modifies databases. Although you can use **installutil** with a variety of subcommands, the syntax described here is specific to the new subcommands that display and enable the feature to store history time stamps in UTC.

The **installutil** command is available on Windows® only.

getutchistorytimestamps

Report whether storing history time stamps in UTC is enabled

Synopsis

```
installutil -please getutchistorytimestamps dbset_name cq_admin_id
cq_admin_pwd [ -database db_name ]
```

Description

The **getutchistorytimestamps** subcommand prints a message that indicates whether storing history time stamps in UTC is enabled.

Options and arguments

-please

Enables the **getutchistorytimestamps** subcommand.

dbset_name

Name of the database set or connection to check.

cq_admin_id

Rational ClearQuest login ID of the administrative user.

cq_admin_pwd

Rational ClearQuest password for the administrative user. To specify a null password, enter an empty set of quotation marks ("").

-database *db_name*

(Optional) Name of the database to check. If omitted, the default value is MASTR, the family name of the schema repository.

Return value

The **installutil** command writes error messages to standard output. A return status of 0 indicates success, non-zero indicates an error.

Examples

- Determine whether UTC history time stamp storage is enabled for database set CQMS.PROD.BOSTON.

```
installutil -please getutchistorytimestamps CQMS.PROD.BOSTON admin secret
```

Starting test getutchistorytimestamps

UTC history timestamps are enabled for dbset CQMS.PROD.BOSTON

Exit code 0 for test getutchistorytimestamps

- Determine whether UTC history time stamp storage is enabled for database LAB1 in the database set CQMS.PROD.BOSTON.

```
installutil -please getutchistorytimestamps CQMS.PROD.BOSTON admin secret -database LAB1
```

Starting test getutchistorytimestamps

UTC history timestamps are enabled for database LAB1 in dbset CQMS.PROD.BOSTON

Exit code 0 for test getutchistorytimestamps

- Determine whether UTC history time stamp storage is enabled for database SAMPL in database set CQMS.PROD.SANFRAN.

```
installutil -please getutchistorytimestamps CQMS.PROD.SANFRAN admin "" -database SAMPL
```

Starting test getutchistorytimestamps

UTC history timestamps are not enabled for dbset CQMS.PROD.SANFRAN

Exit code 0 for test getutchistorytimestamps

setutchistorytimestamps

Enables storing history time stamps in UTC

Synopsis

```
installutil -please setutchistorytimestamps dbset_name cq_admin_id cq_admin_pwd
```

Description

The **setutchistorytimestamps** subcommand enables UTC history time stamp storage in the MASTR database and all user databases at the working master site. This feature is enabled at other sites when they synchronize with the working master site.

Restrictions

- You must have Rational ClearQuest super user privileges to run this subcommand.
- You must run this subcommand at the working master site.

Options and arguments

-please

Enables the **setutchistorytimestamps** subcommand.

dbset_name

Name of the database set or connection to convert to UTC.

cq_admin_id

Rational ClearQuest login ID of the administrative user.

cq_admin_pwd

Rational ClearQuest password for the administrative user. To specify a null password, enter an empty set of quotation marks ("").

Return value

The **installutil** command writes error messages to standard output. A return status of 0 indicates success, non-zero indicates an error.

Example

- Enable storing history time stamps in UTC for database set CQMS.PROD.SANFRAN.
installutil -please setutchistorytimestamps CQMS.PROD.SANFRAN
 Starting test setutchistorytimestamps
 Exit code 0 for test setutchistorytimestamps

ResultSet object methods

This section describes two new API methods of the ResultSet object that control whether dates returned in query, report, and chart results are converted to client local time for display or are displayed as read from the database.

- "GetConvertToLocalTime"
- "SetConvertToLocalTime" on page 12

GetConvertToLocalTime

Description

The **GetConvertToLocalTime** method returns the convert-to-local-time property of the ResultSet object. When this property is True, UTC database history dates are converted to client local time for version 7.0 clients to display in query, report, and chart results. When this property is False, UTC database history dates are not converted to client local time for display. The default property setting is True.

Syntax

Perl

```
$resultset->GetConvertToLocalTime();
```

Identifier	Description
<i>resultset</i>	A ResultSet object representing the rows and columns of data resulting from a query.

Identifier	Description
<i>Return value</i>	Returns a Boolean True if the convert-to-local-time property is set; False otherwise.

SetConvertToLocalTime

Description

The **SetConvertToLocalTime** method of the ResultSet object enables or disables the convert-to-local-time property of the ResultSet object. When this property is True, UTC database history dates are converted to client local time for version 7.0 clients to display in query, report, and chart results. When this property is False, UTC database history dates are not converted to client local time for display. Instead, version 7.0 clients display dates in UTC as these are stored on the server.

Syntax

Perl

```
$resultset->SetConvertToLocalTime( value );
```

Identifier	Description
<i>resultset</i>	A ResultSet object representing the rows and columns of data resulting from a query.
<i>value</i>	A Boolean True enables conversion of UTC-stored database history dates to client local time for Rational ClearQuest version 7.0 clients to display in query, report, and chart results. A Boolean False disables conversion of UTC-stored database history dates to client local time for display. Instead, version 7.0 clients display dates in UTC as these are stored on the server.
<i>Return value</i>	None.

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:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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:

IBM Corporation
 Department BCFB
 20 Maguire Road
 Lexington, MA 02421
 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 measurement 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

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

(c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

Additional legal notices are described in the legal_information.html file that is included in your Rational software installation.

Trademarks

IBM, ClearQuest, and Rational are trademarks of International Business Machines Corporation 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.



IBM Confidential
Printed in USA