precisely

# Visibility API

Installation Guide
For Windows, Unix, and Linux

# Table of Contents


Introduction ..... 4
About This Installation Guide ..... 4
Visibility API Environment ..... 5
Functional Overview ..... 6
Installation Prerequisites ..... 7
Installation Overview ..... 7
Contacting Customer Support ..... 7
Installing on Windows ..... 8
Install the Visibility API ..... 8
Configure the Visibility API ..... 9
Implement the Visibility API ..... 10
What's Next? ..... 10
Installing on UNIX or Linux ..... 11
Install the Visibility API ..... 11
Configure the Visibility API ..... 12
Implement the Visibility API ..... 13
What's Next? ..... 14
Troubleshooting ..... 15
Run the Recovery Script ..... 15
Troubleshooting Tips ..... 16
Configuration Settings ..... 18
Visibility API Properties ..... 18
log4j Properties ..... 22

# Introduction

This chapter introduces you to the Visibility API and this installation guide in the following sections:

- "About This Installation Guide"
- "Visibility API Environment"
- "Functional Overview"
- "Installation Prerequisites"
- "Installation Overview"
- "Contacting Customer Support"

Along with this guide, you should have downloaded the Visibility API installation files.

## About This Installation Guide

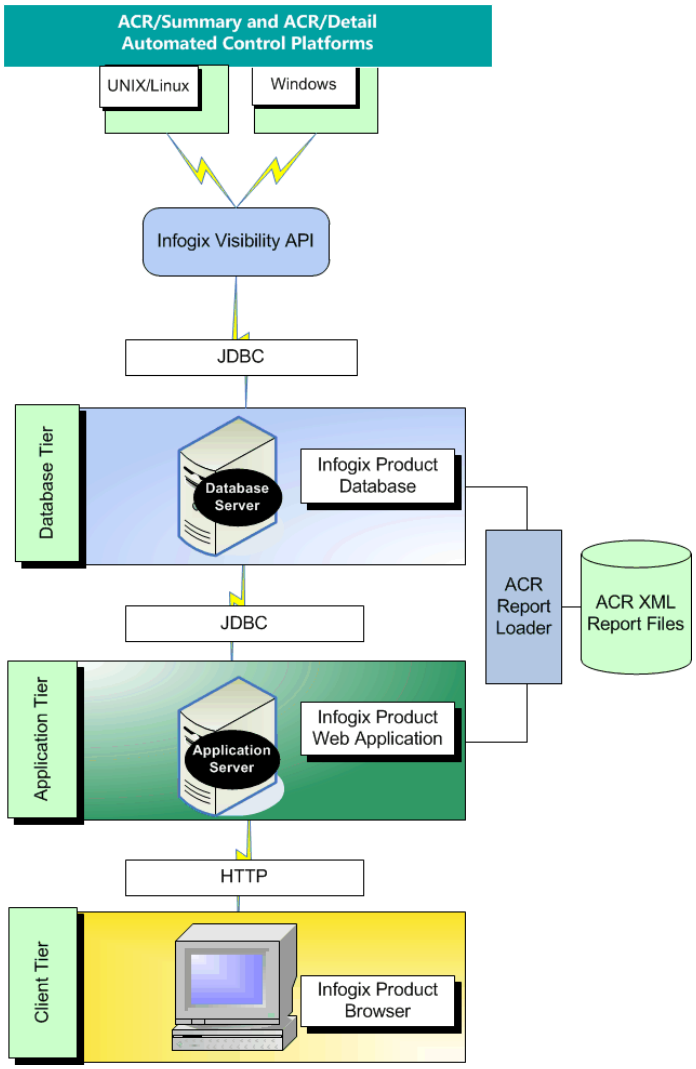
This guide outlines the steps necessary to install the Visibility API for use with Insight® or Perceive®.

After this introduction, the guide has the following chapters:

| Chapter | Description |
|---|---|
| Chapter 2, "Installing on Windows" | Explains how to install and configure the Precisely Visibility API on Windows®. |
| Chapter 3, "Installing on UNIX or Linux" | Explains how to install and configure the Visibility API on UNIX® or Linux®. |
| Chapter 4, "Troubleshooting" | Provides instructions for the recovery utility and includes troubleshooting tips. |
| Appendix A, "Configuration Settings" | Provides the information to configure properties for the Visibility API and set the environment variables. |

# Visibility API Environment

The following diagram illustrates the architectural environment of the Visibility API. To send control information and results to Insight® or Perceive®, install the Visibility API on each machine where ACR® is installed.




See "Functional Overview".

# Functional Overview

The Visibility API must be installed on the same server where ACR is installed. This product is the gateway that bridges the ACR product with Insight or Perceive, or both.

This section describes the type of information that ACR sends to its target applications, Insight and Perceive:

- Control definition information
  - Update messages
  - Save messages
- Control execution information
  - Start-of-job messages
  - End-of-job messages
  - Report messages

The options that you set in ACR/Summary and ACR/Detail and the properties that you configure in the Visibility API determine the type of ACR control information to store in the target application. If report auto-loading is enabled in the target application, the API automatically loads the data.

After the ACR control information is sent to the target application, administrative users can view the data in the user interface through a Web browser.

## From ACR to Insight

In Insight, the ACR control information is stored in the Insight database.

Administrators and View Builders can set up control groups and control views for Control Viewers, and Auditors can view control definition, control execution, and report events in the audit trail.

## From ACR to Perceive

In Perceive, the ACR control information is stored in the Job Repository.

Perceive DataEntity Admins and DataView Admins can set up job data entities and data views for Report Writers to use to create reports for Report Viewers.

# Installation Prerequisites

To install release of the Visibility API, refer to the required applications, listed below. The versions listed are the minimum requirements for each application:

- If you use **Insight, release** (which uses an **Oracle®** or **SQL Server® database**)
- If you use **Perceive, release** (which uses an **Oracle** or **SQL Server database**)
- **ACR/Summary®** or **ACR/Detail® ,** or both
- **Java™ version 1.6.0 or higher (32 bit)**

# Installation Overview

Below is a high-level summary of the process you will follow.

1. Review the prerequisites, listed above.
2. Proceed with installation for the Visibility API on Windows, UNIX, or Linux where ACR is installed. Instructions for Windows are in Chapter 2. Instructions for UNIX or Linux are in Chapter 3.
3. Follow the instructions in the appropriate chapter to configure the Visibility API.
4. Follow the instructions in the appropriate chapter to implement the Visibility API.

# Contacting Customer Support

If you need assistance, contact Precisely Customer Support at support.precisely.com

# Installing on Windows

This chapter provides instructions to install, configure, and implement the Visibility API on Windows for use with Insight or Perceive with an Oracle or SQL Server database.

This chapter includes the following sections:

- "Install the Visibility API"
- "Configure the Visibility API"
- "Implement the Visibility API"
- "What's Next?"

Review each section and carefully follow all the steps to ensure correct setup and installation.

## Install the Visibility API

These instructions assume that you have satisfied the prerequisites, as described in "Installation Prerequisites".

The machine where you install the Visibility API must have a JDBC driver connection to the target application database. Complete the following steps to install the Visibility API.

1. Insert the Visibility API installation CD into the CD-ROM drive of the installation machine.
2. From the installation CD, extract the contents of the Windows/VisibilityAPI.zip file to an installation directory of your choice.

   This process creates an Precisely directory with the following directory structure:

   - VisAPI—this subdirectory contains the config, bin, lib, and logs subdirectories.
     - config—this subdirectory contains the following files:
       - Config.properties—configuration properties for the Visibility API and its connection to the Insight database, or the Job repository in Perceive.
       - log4j.properties—configuration file for the logging process.

  - timezone.properties—lists all the available time zone values.
- bin—this subdirectory contains the APIenv.bat and recover.bat scripts. APIenv.bat displays the correct information needed to manually set the system variables correctly. The recover.bat script can perform the recovery of ACR control information in the event of a communication failure with the Insight database or the Perceive Job Repository.
- lib—this subdirectory contains the necessary jdbc drivers for Oracle and SQL Server and all the JAR files, including the VisibilityAPI.jar file.
- logs—api.log files will be stored in this subdirectory when they are produced.

# Configure the Visibility API

The configuration process includes the modification of properties in the <install_dir>/Precisely/VisAPI/config/Config.properties and <install_dir>/Precisely/VisAPI/config/log4j.properties files. See "Configuration Settings" for detailed information.

1. Configure the properties in Config.properties.
   See "Visibility API Properties" for more information.

   If you have multiple target applications, you will specify multiple values for the SUBSCRIBERS property.
   For example, if you use the Visibility API with both Insight and Perceive, you might have the following values for the SUBSCRIBERS property: SUBSCRIBERS=INSIGHT,PERCEIVE

   For those properties preceded by the subscribers value, enter a separate property for each target application. For example:
   INSIGHT.DBUSER=username1
   PERCEIVE.DBUSER=username2

2. Configure the properties in log4j.properties.
   See "log4j Properties" for more information.

# Implement the Visibility API

After you have installed and configured the Visibility API, you will need to implement the API by setting system environment variables.
Although you must set these values manually, you can run the APIenv.bat script to obtain the values you will need to use.

1. Run the APIenv.bat script. It will display the system environment variables and values that you need to be set.
   a. Navigate to <install_dir>/Precisely/VisAPI/bin.
   b. From a command line, run the following script:

      ```
      APIenv.bat
      ```

      The script displays the values to use for the following system variables:
      - CONTROLSINSIGHT_HOME
      - CLASSPATH
      - INSGTDBG

      The script will inform you if your JAVA_HOME is not set. You must correct this to complete the implementation.
2. Set your system environment variables to the values displayed. The values may be copied and pasted.

# What's Next?

If there are multiple installations of ACR, the Visibility API will need to be installed with each ACR installation that will be used with the Visibility Products.

# Installing on UNIX or Linux

This chapter describes how to install, implement, and configure the Visibility API on UNIX (AIX® or Oracle Solaris®) or Linux (Red Hat) for use with Insight or Perceive which use an Oracle or SQL Server database.

This chapter includes the following sections:


- "Install the Visibility API"
- "Configure the Visibility API"
- "Implement the Visibility API"
- "What's Next?"


## Install the Visibility API

These instructions assume that you have satisfied the prerequisites, as described in "Installation Prerequisites".

The machine where you install the Visibility API must have a JDBC driver connection to the target application database. Complete the following steps to install the Visibility API.

1. Insert the Visibility API installation CD into the CD-ROM drive of the installation machine.
2. From the installation CD, extract the contents of the appropriate tar file into an installation directory of your choice:

   ```
   Unix/VisibilityAPI.tar
   ```

   or

   ```
   Linux/VisibilityAPI.tar
   ```

   For example, tar -xvpf VisibilityAPI.tar <install_dir>.

   This process creates an Precisely directory with the following directory structure:

   - VisAPI—this subdirectory contains the config, bin, lib, and logs subdirectories.
     - config—this subdirectory contains the following files:
       - Config.properties—configuration properties for the Visibility API and its connection to the Insight database, or the Job repository in Perceive.

- log4j.properties—configuration file for the logging process.
- timezone.properties—lists all the available time zone values.

- bin—this subdirectory contains the following scripts:
  - APIenv.sh—this script sets the Visibility API environment variables for UNIX.
  - recover.sh—a Visibility API script that recovers ACR control information in the event of a communication failure with the Insight database or the Job Repository in Perceive.
- lib—this subdirectory contains the necessary JDBC drivers for Oracle and SQL Server and all the JAR files, including the VisibilityAPI.jar file.
- logs—api.log files will be stored in this subdirectory when they are produced.

3. Make sure that the owner of the ACR Summary and ACR Detail server has read, write, and execute permissions for the Visibility API files, and read and write access to the config directory and logs directory.

# Configure the Visibility API

The configuration process includes the modification of properties in the <install_dir>/Precisely/VisAPI/config/Config.properties and <install_dir>/Precisely/VisAPI/config/log4j.properties files. See Appendix A, "Configuration Settings" for detailed information.

1. Configure the properties in Config.properties. See "Precisely Visibility API Properties" for more information.
   If you have multiple target applications, you will specify multiple values for the SUBSCRIBERS property.
   For example, if you use the Visibility API with both Insight and Perceive, you might have the following values for the SUBSCRIBERS property: SUBSCRIBERS=INSIGHT,PERCEIVE

For those properties preceded by the subscribers value, enter a separate property for each target application. For example:
INSIGHT.DBUSER=username1
PERCEIVE.DBUSER=username2

2 Configure the properties in log4j.properties. See "log4j Properties" for more information.

# Implement the Visibility API

After you have installed and configured the Visibility API, you will need to perform additional tasks to implement the API. The tasks will be to modify the ACR environment file, set the COBOL driver, and restart the server.

## Modify the ACR Environment File

To ensure that the ACR/Summary and ACR/Detail processing environment uses the Visibility API environment settings, complete the following steps.

1. Navigate to <ACR_install_dir>/bin.
2. Add the following lines in the ACR environment file, acrprf, at the end of the ACR/ Summary and ACR/Detail Section in the file:

   ```
   export INSGTDBG=Y
   ```

   INSGTDBG is an optional setting. See "Display Debug Messages in SYSOUT" for more information.

   ```
   export JAVA_HOME=<jdk6_install_dir>
   ```

   where <jdk6_install_dir> is the location of the Java installation that will be used by the Visibility API.

   ```
   export API_INSTALLDIR=<install_dir>/Precisely/VisAPI
   ```

   where <install_dir> is the location where the Visibility API has been installed.

   ```
   . $API_INSTALLDIR/bin/APIenv.sh
   ```

---

**Note**: Ensure that the user account, under which the ACR/Summary and ACR/Detail process is running, has write permission to the Visibility API installation directory, and its subdirectories.

---

## Set the COBOL Driver

To ensure that the ACR/Summary and ACR/Detail processing environment uses the proper Micro Focus® driver with the Visibility API, you must set the COBOL driver.

1. Navigate to <ACR_install_dir>/sumdetsd.ini
2. Edit the DRIVER setting as follows:

```
DRIVER=cobjrun
```

## Restart the Server

Stop and then restart the ACR/Summary and ACR/Detail server. This step implements the changes that you made in "Modify ACR Environment Files" and "Set the COBOL Driver". Stopping and restarting the server is necessary if you intend to use ACR/Workbench with a UNIX configuration.

# What's Next?

If there are multiple installations of ACR, the Visibility API will need to be installed with each ACR installation that will be used with the Visibility Products.

# Troubleshooting

This chapter contains instructions for running the recovery utility and other troubleshooting tips. This chapter includes the following sections:


- "Run the Recovery Script"
- "Troubleshooting Tips"


## Run the Recovery Script

If the Visibility API is configured for message recovery when communication with a Visibility Product fails, control execution and report messages are saved to an InsightRecovery.log and a staging directory.

Control execution messages are saved in the following location:

<install_dir>/Precisely/VisAPI/config/InsightRecovery.log

Report messages are saved in the following location:

<install_dir>/Precisely/VisAPI/config/recoverystaging

After the messages are saved, the Recovery script can be used to send the recovered messages to the Visibility Product where communication failed. You can run this script as often as needed.

Complete the following steps to run the recovery script:

1. Navigate to <install_dir>/Precisely/VisAPI/bin
2. Run the following command to execute the script:

   ```
   ./recover.sh
   ```

   JAVA_HOME must be set for this script to run.

   The script has run successfully when it returns to the command prompt and there is no error or warning message displayed.

   If there is a warning or error message, the execution information has been rewritten to the InsightRecovery.log and report information has been retained. Please run the Recovery script again when communication has been reestablished.

# Troubleshooting Tips

These troubleshooting procedures may be helpful when expected ACR data cannot be seen in a Visibility Product.

## Display Debug Messages in SYSOUT

Visibility API DEBUG messages can be displayed in the ACR control's SYSOUT.

### For Windows

Set the INSGTDBG environment variable to Y. See "Implement the Visibility API".

### For UNIX or Linux

1. Navigate to <ACR_install_dir>/bin/acrprf
2. Add "export INSGTDBG=Y" near the end of the acrprf file.

This setting generates and displays debug messages in the ACR control's SYSOUT during Visibility API processing.

## Display Debug Messages in the API Log

The Visibility API can produce diagnostic logging information in <install_dir>Precisely/VisAPI/logs/api.log depending on the logging level set in log4j.properties.

1. Navigate to <install_dir>/Precisely/VisAPI/config/log4j.properties.
2. Change log4j.rootLogger=INFO, apilog to log4j.rootLogger=DEBUG, apilog

A log level of "DEBUG" will display information about the processing of the API and will also display the contents of the messages that are being sent to the Visibility Products.

## Resolve Out of Memory Errors

Some ACR reports can be very large. Sending the XML report file can result in the following error:

"java.lang.OutOfMemoryError: Java heap space"

If this happens, you may need to perform one of the following tasks:

- Increase the size of the total JVM memory
- Modify the ACR control so that the generated reports are smaller

# Configuration Settings

This appendix provides additional information about the names and values used to configure the Visibility API on Windows, UNIX, or Linux.

## Visibility API Properties

The table that follows describes the configuration properties for the Visibility API. These properties are found in the <install_dir>/Precisely/VisAPI/config/Config.properties file.

| Visibility API Property | Description |
|---|---|
| PLATFORMID | Optional. If a value is specified for this property, it is appended to every ACR® Job/Step/Qualifier control name that runs using this API. It can be used to uniquely identify ACR® controls that have the same name but run on different instances of ACR® and the Visibility API. For example, if "SAMPLExxSTEP1xxx01" runs on both DEV and PROD and those names were defined as the PLATFORMID value in the associated Config.properties file, one control would be named "SAMPLExxSTEP1xxx01DEV" and the other would be named "SAMPLExxSTEP1xxx01PROD". |
| TIMEZONE | This property should be set to the time zone where corresponding ACR® controls run. It is used to store the correct start and end times of each execution in the Visibility Product database.<br>For a list of valid time zone values, see the timezone.properties file in <installdir>/Precisely/VisAPI/config. Note that values are case-sensitive and do not include spaces. |
| DEBUG | This property's value is set to "N" meaning that ACR® execution information is sent to Visibility Product databases by default. If it is set to "X" instead, ACR® will process as usual but Visibility Product databases will not be updated with execution information. |

***Visibility API Properties***

| Visibility API Property | Description |
|---|---|
| SUBSCRIBERS | Specify a value to identify the subscribing or target application for the properties that follow.<br>Use the default value, if applicable, or specify a user-defined value.<br>This property can have multiple values. Separate each value with a comma. For example, SUBSCRIBERS=INSIGHT,PERCEIVE |
| *subscribername*.AUTODETECT | Precede this property with the value defined for the SUBSCRIBERS property. Specify Y to recognize and send controls that run on the source platform but do not yet exist on the target application. Otherwise, specify N to have the target application ignore these controls.<br>Add a separate property for each target application. |
| *subscribername*.COMMMODULE | Precede this property with the value defined for the SUBSCRIBERS property. Add a separate property for each target application.<br>Leave the value for this property blank.<br>Only z/OS uses a communication module. The system references the value specified in the configuration for the communication method. |
| *subscribername*.DSNAME | Precede this property with the value defined for the SUBSCRIBERS property. Specify either a JNDI name or specify the values for the subscribername.DBDRIVER, subscribername.DBURL, subscribername.DBUSER, and subscriber name.DBPASSWORD properties.<br>Add a separate property for each target application. |
| *subscribername*.DBDRIVER | Precede this property with the value defined for the SUBSCRIBERS property. Specify the class name of the database driver for the target application database.<br>Add a separate property for each target application. |
| *subscribername*.DBURL | Precede this property with the value defined for the SUBSCRIBERS property. Specify the complete URL for the target application database. Add a separate property for each target application. |

| Visibility API Property | Description |
|---|---|
| *subscribername*.DBUSER | Precede this property with the value defined for the SUBSCRIBERS property. Specify the user name for the user who is authorized to connect to the target application database.<br>Add a separate property for each target application. |
| *subscribername*.DBPASSWORD | Precede this property with the value defined for the SUBSCRIBERS property. Specify the password for the username in the previous property, subscribername.DBUSER.<br>Add a separate property for each target application. |
| *subscribername*.ACRS_SOJ | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Summary start-of-job messages: R = send the message with a recovery option; Y = send the message; N = do not send the message.<br>Add a separate property for each target application. |
| *subscribername*.ACRS_EOJ | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Summary end-of-job messages: R = send the message with a recovery option; Y = send the message; N = do not send the message.<br>Add a separate property for each target application. |
| *subscribername*.ACRS_CRPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Summary control reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |
| *subscribername*.ACRS_URPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Summary user reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |

***Visibility API Properties***

| Visibility API Property | Description |
|---|---|
| *subscribername*.ACRS_FRPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Summary free-form reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |
| *subscribername*.ACRD_SOJ | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Detail start-of-job messages: R = send the message with a recovery option; Y = send the message; N = do not send the message.<br>Add a separate property for each target application. |
| *subscribername*.ACRD_EOJ | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Detail end-of-job messages: R = send the message with a recovery option; Y = send the message; N = do not send the message.<br>Add a separate property for each target application. |
| *subscribername*.ACRD_CRPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Detail control reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |
| *subscribername*.ACRD_URPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Detail user reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |

| Visibility API Property | Description |
|---|---|
| *subscribername*.ACRD_FRPT | Precede this property with the value defined for the SUBSCRIBERS property. Specify how to handle ACR/Detail free-form reports: R = send the report with a recovery option; Y = send the report; N = do not send the report.<br>Add a separate property for each target application. |
| DATEFORMAT | Accept the default format, YYMMDD. |
| DATESEPARATOR | Accept the default separator, a forward slash (/). |
| PARSETYPE | Accept the default type, sc. |
| JOB1STEP1Q1 | Accept the default setting. |

# log4j Properties

The table that follows describes the log4j properties for the Visibility API. These properties are found in the <install_dir>/Precisely/VisAPI/configlog4j.properties file.

| log4j Property | Description |
|---|---|
| log4j.rootLogger | Specify the logging level at which to log information. The options include DEBUG, INFO, WARN, ERROR, FATAL and OFF.<br>Use the following format: INFO, apilog |
| log4j.appender.console | Accept the default value for this property, which sets the console to be a console appender. |
| log4j.appender.console.layout | Accept the default value for this property, which sets the console to be a console appender. |
| log4j.appender.console.layout.<br>ConversionPattern | Accept the default value for this property, which sets the console to be a console appender. |
| log4j.appender.apilog | Accept the default value for this property, which sets the API log to be a rolling file appender. |
| log4j.appender.apilog.layout | Accept the default value for this property, which sets the API log to be a rolling file appender. |
| log4j.appender.apilog.layout.<br>ConversionPattern | Accept the default value for this property, which sets the API log to be a rolling file appender. |

***log4j Properties***

| log4j Property | Description |
|---|---|
| log4j.appender.apilog.File | Specify the full path, including the file name for the log file. |
| log4j.appender.apilog.MaxBackupIndex | Specify the number of logs to retain in the logs subdirectory. |
| log4j.appender.apilog.MaxFileSize | Specify the maximum size in KB, MB, or GB for the log file. When the contents of the file reach the maximum size, the log file is rolled over and a new file is created. |

## Notices

### Trademarks

Infogix, the Infogix logo, ACR, ACR/Detail, ACR/Summary, ACR/Workbench, ACR/Connector, Infogix Assure, Infogix Insight, ACR/Instream, ACR/File, Infogix ER, Infogix Perceive, Data3Sixty, and Data360 are registered trademarks of Precisely. Data3Sixty Analyze, Data3Sixty Govern, Data3Sixty DQ+, Data360 Analyze, Data360 Govern and Data360 DQ+ are trademarks of Precisely. Any other trademarks or registered trademarks are the property of their respective owners.

1700 District Ave Ste 300
Burlington MA 01803-5231
USA

www.precisely.com

Copyright 2005, 2024 Precisely