



i-net Crystal-Clear 8

Migration Guide

from previous versions to version 8

Table of Contents

1 Content	4
2 Java Version	5
3 API Changes	6
3.1 Added Classes.....	6
3.2 Added Methods.....	6
3.3 Methods and Constants deprecated as of Version 8.....	7
3.4 Moved Methods.....	9
3.5 Removed deprecated Methods and Constants.....	9
4 i-net Crystal-Clear Configuration in Version 8	11
4.1 Preferences instead of crystalclear.properties.....	11
4.2 Importing old crystalclear.properties.....	11
4.3 Scopes.....	12
4.3.1 System Scope.....	12
4.3.2 User Scope.....	12
4.3.3 Temporary Scope.....	13
4.4 Choosing a configuration to use.....	13
4.4.1 ...in i-net DesignerXML.....	13
4.4.2 ...for the Report Server (Listener).....	13
4.4.3 ...for i-net Crystal-Clear running as Servlet.....	14
4.5 Changing Configurations “On-the-fly”.....	15
4.5.1 Configuration Manager Tool.....	15
4.5.2 API.....	15
5 Datasources and Configuration of Driver DLL's	16
5.1 GUI.....	16
5.2 Old Reports and Crystal Reports Templates.....	16
5.3 API.....	16
6 Classic Viewer removed	17
7 JAR files for installation	18
8 New Chart API	19
9 Troubleshooting / FAQ	20

9.1 I don't see a crystalclear.properties file in our installation directory! How do I configure my i-net Crystal-Clear installation?.....20

9.2 I have entered a valid license key but still am receiving the error “No license key” or “No configuration found”!.....20

9.3 I don't see any database dll properties in the new configuration dialog! How do I configure my database connections?.....20

9.4 I have an application still using the old Chart API - using these API methods now causes an error! Do I have to change all of my code to the new API?.....21

1 Content

The following document describes the various things to know when migrating to version 8.x of i-net Crystal-Clear. Version 8.x included various restructuring that might lead to minor incompatibilities with older applications.

2 Java Version

i-net Crystal-Clear and i-net DesignerXML version 8.x require at least Java 5. If your application runs with an older version of Java (version 1.4.2 or earlier), you should migrate your application to Java 5 before migrating to version 8 of i-net Crystal-Clear.

3 API Changes

3.1 Added Classes

- ▷ com.inet.report.config
 - ▷ Configuration
 - ▷ ConfigurationManager
 - ▷ LicenseKeyInfo
- ▷ com.inet.report
 - ▷ Chart2
- ▷ com.inet.report.chart
 - ▷ All classes in this package and in it's sub packages.

3.2 Added Methods

- ▷ com.inet.report.cache.Cache
 - ▷ getFailureCause(ReportCacheKey)
- ▷ com.inet.report.Engine.getStatus()
- ▷ com.inet.report.FieldElement
 - ▷ getBaseUrl()
 - ▷ setBaseUrl(String)
 - ▷ getBaseUrlFormula()
 - ▷ setBaseUrlFormula(FormulaField)
- ▷ com.inet.report.ParagraphProperties
 - ▷ getFirstLineIndentFormula()
 - ▷ setFirstLineIndentFormula(FormulaField)
 - ▷ getLeftIndentFormula()
 - ▷ setLeftIndentFormula(FormulaField)
 - ▷ getLineSpacingAbsoluteFormula()
 - ▷ setLineSpacingAbsoluteFormula(FormulaField)
 - ▷ getLineSpacingRelativeFormula()
 - ▷ setLineSpacingRelativeFormula(FormulaField)
 - ▷ getRightIndentFormula()
 - ▷ setRightIndentFormula(FormulaField)
- ▷ com.inet.report.ReportProperties
 - ▷ getReportLocale()
 - ▷ setReportLocale(java.util.Locale)
- ▷ com.inet.report.Section
 - ▷ addChart2(ChartStyle,int,int,int,int)

- ▷ com.inet.report.TextProperties
 - ▷ getGlyphOrientationFormula()
 - ▷ setGlyphOrientationFormula(FormulaField)
 - ▷ getTextRotationFormula()
 - ▷ setTextRotationFormula(FormulaField)

- ▷ com.inet.report.ValueProperties
 - ▷ getDateFormatType()
 - ▷ setDateFormatType(int)
 - ▷ getDateFormatTypeFormula()
 - ▷ setDateFormatTypeFormula(FormulaField)
 - ▷ getTimeFormatType()
 - ▷ setTimeFormatType(int)
 - ▷ getTimeFormatTypeFormula()
 - ▷ setTimeFormatTypeFormula(FormulaField)

- ▷ com.inet.viewer.ReportView
 - ▷ getDefaultAttributeSet(int,int)
 - ▷ print(PrinterJob,PrintRequestAttributeSet)

- ▷ com.inet.viewer.SwingReportView
 - ▷ showPrintDialog(Component,PrinterJob,HashPrintRequestAttributeSet)

- ▷ com.inet.viewer.SwingReportViewer
 - ▷ getPrinterDefaultFormatHandling()
 - ▷ setPrinterDefaultFormatHandling(int)

3.3 Methods and Constants deprecated as of Version 8

The following list includes the methods that are deprecated in version 8 and should no longer be used. Please consult the Javadoc of the individual methods for information on which newer methods to use instead.

- ▷ com.inet.designer.DesignerDataModel
 - ▷ openChartDialog(Chart)

- ▷ com.inet.report.AbstractValueElement
 - ▷ getDateWindowsDefaultType()
 - ▷ setDateWindowsDefaultType(int)
 - ▷ getDateWindowsDefaultTypeFormula()
 - ▷ setDateWindowsDefaultTypeFormula(FormulaField)
 - ▷ setUseSystemDefaultFormat(boolean)
 - ▷ getUseSystemDefaultFormat()
 - ▷ setUseSystemDefaultFormatFormula(FormulaField)
 - ▷ getUseSystemDefaultFormatFormula()

- ▷ com.inet.report.Chart
 - ▷ all members of this class
- ▷ com.inet.report.ChartFormatProperties
 - ▷ getDateWindowsDefaultType()
 - ▷ setDateWindowsDefaultType(int)
 - ▷ getDateWindowsDefaultTypeFormula()
 - ▷ setDateWindowsDefaultTypeFormula(FormulaField)
 - ▷ setUseSystemDefaultFormat(boolean)
 - ▷ getUseSystemDefaultFormat()
 - ▷ setUseSystemDefaultFormatFormula(FormulaField)
 - ▷ getUseSystemDefaultFormatFormula()
- ▷ DatabaseTables
 - ▷ createDatasource(String,String,String,String,String,String)
 - ▷ createDatasource(String,String,String,String)
- ▷ com.inet.report.Datasource
 - ▷ dll_connectionByConfiguration
 - ▷ getCatalog()
 - ▷ setCatalog(String)
 - ▷ getDll()
 - ▷ setDll(String)
 - ▷ getHost()
 - ▷ setHost(String)
 - ▷ getUsername()
 - ▷ setUsername(String)
 - ▷ getURL()
 - ▷ setUrl(String)
- ▷ com.inet.report.Element
 - ▷ getHyperlinkType()
 - ▷ setHyperlinkType(int)
- ▷ com.inet.report.Engine
 - ▷ setPath(String, String, String)
 - ▷ setPath(java.net.URL)
 - ▷ getGroupTree(int, int)
- ▷ com.inet.report.GroupTree
 - ▷ getData(byte[],int,int)
 - ▷ createErrorTree(boolean)
- ▷ com.inet.report.HyperlinkProperties
 - ▷ NO_HYPERLINK
 - ▷ FILE_HYPERLINK
 - ▷ WEBSITE_HYPERLINK
 - ▷ EMAIL_HYPERLINK
 - ▷ FIELD_VALUE_HYPERLINK

- ▷ getHyperlinkType()
- ▷ setHyperlinkType(int)

- ▷ com.inet.report.RDC
 - ▷ getCrystalClearPropertyPath()
 - ▷ setConfigurationProperties(Properties)
 - ▷ getConfigurationProperties()

- ▷ com.inet.report.Section
 - ▷ addChart(int, int, int, int, int)

- ▷ com.inet.report.ValueProperties
 - ▷ USE_WINDOWS_LONG_DATE
 - ▷ USE_WINDOWS_SHORT_DATE
 - ▷ NOT_USING_WINDOWS_DEFAULTS
 - ▷ getDateWindowsDefaultType()
 - ▷ setDateWindowsDefaultType(int)
 - ▷ getDateWindowsDefaultTypeFormula()
 - ▷ setDateWindowsDefaultTypeFormula(FormulaField)
 - ▷ setUseSystemDefaultFormat(boolean)
 - ▷ getUseSystemDefaultFormat()
 - ▷ setUseSystemDefaultFormatFormula(FormulaField)
 - ▷ getUseSystemDefaultFormatFormula()

- ▷ com.inet.viewer.SwingViewerContext
 - ▷ SwingViewerContext()

3.4 Moved Methods

- ▷ com.inet.report.Element
 - ▷ getManualFormatFormula() moved to AbstractValueElement.getManualFormatFormula()
 - ▷ setManualFormatFormula(FormulaField) moved to AbstractValueElement.setManualFormatFormula(FormulaField)

3.5 Removed deprecated Methods and Constants

The following methods had been deprecated for longer than 2 major versions and have now been entirely removed in version 8 of i-net Crystal-Clear.

- ▷ com.inet.report.Area
 - ▷ setSectionWillBePrinted(int)
 - ▷ getSectionWillBePrinted()

- ▷ com.inet.report.Chart
 - ▷ setXAxisSource(Field)
 - ▷ getXAxisSource()

- ▷
- ▷ com.inet.report.CrossTab
 - ▷ dropColumn()
- ▷ com.inet.report.DatabaseTables
 - ▷ *_JOIN
 - ▷ addJoin(String , String , String , String , int)
 - ▷
- ▷ com.inet.report.Engine
 - ▷ setPath(String)
 - ▷ setPath(String, String)
- ▷ com.inet.report.Fields
 - ▷ getGroupSortField(int)
 - ▷ getGroupSortFieldsCount()
- ▷ com.inet.report.FormulaField
 - ▷ getBasicSyntax()
 - ▷ setBasicSyntax(boolean)
- ▷ com.inet.report.Group
 - ▷ setKeepGroupTogether(boolean)
 - ▷ getKeepGroupTogether()
- ▷ com.inet.report.RDC
 - ▷ gc(Engine)
- ▷ com.inet.report.ReportProperties
 - ▷ getPageWidth()
 - ▷ setPageWidth(int)
- ▷ com.inet.report.ReportServletJSP
 - ▷ props

4 i-net Crystal-Clear Configuration in Version 8

4.1 Preferences instead of crystalclear.properties

Starting in version 8 of i-net Crystal-Clear, `crystalclear.properties` are no longer used. All configurations are instead stored in the operating system's preferences - where these are located depends on the operating system. In Microsoft Windows, i-net Crystal-Clear uses the Windows registry, while on Unix systems the preferences are stored in the file system (`/etc/.java`, or `~/.java`). Because of this change, it is now possible to comfortably store and manage configurations in a central place, you no longer have to work with various `crystalclear.properties` files.

4.2 Importing old `crystalclear.properties`

All existing `crystalclear.properties` files can be imported into a preferences scope (System, User, Temporary) with the new Configuration Manager. Note that if i-net Crystal-Clear will be started as a service, it is recommended to use the System scope since the i-net Crystal-Clear service will most likely be running as a different user.

When importing a `crystalclear.properties` file, its settings are completely copied into the configuration. The database driver DLL-properties, however, must first be converted into data sources in order for them to be used in version 8. An additional dialog will show the choices for converting these properties and will offer a choice of data source configuration per database driver DLL. If no database driver DLLs are to be imported, the dialog can simply be canceled, otherwise you can skip the database driver DLLs you do not need by choosing “<do not import>”. Also, for each data source Text body, an individual scope can be set (System, User, Temporary), depending on the current read and write permissions. A context menu additionally offers help for the simple choosing of drivers and scopes. Please note that after this conversion of the properties, imported configurations do not contain the database driver DLL properties any more.

If you do not want (or are not able) to use the Configuration Manager GUI for

importing your `crystalclear.properties` files, you can do the same via the command line. The program argument `-importConfig` will start the import process without displaying any window.

```
java -jar CrystalClear.jar -importConfig <name> <file>
```

In the argument `<name>` you must specify the scope and name of the configuration that is to be created, e.g. “System/Default” or “User/myConfig”. Only scopes “System” and “User” are allowed.

The argument `<file>` must contain the path to a readable and valid properties file, e.g. `crystalclear.properties`.

Please note that the import by command line will not convert any DLL-properties to data sources. This can only be done by using the Configuration Manager.

4.3 Scopes

You can store your configurations in one of three different scopes: the System scope, the User Scope, and the Temporary scope. What this means “behind the scenes” for the storage will depend on the operating system your platform uses.

4.3.1 System Scope

The System scope is viewable by all users who have read access to system properties. On many systems, the System scope is only writable by administrators or users with system privileges. Typical uses of this scope would be for deployments on application servers which run as a different user than the normal user, and therefore could not see configurations stored in the User scope. It's also conceivable to store a system-wide scope for various different users in order to not have to create and configure multiple configurations for multiple users.

4.3.2 User Scope

The User Scope is only visible to the user it belongs to. This means configurations stored in the User scope by user A will not be visible to user B. This scope is useful for situations where there is only one user on the system, since the user scope is almost always writable and readable by that user.

4.3.3 Temporary Scope

Storing configurations in this scope will cause the configuration to only work as long as the session runs. As soon as the Java VM is terminated, the configuration will be lost. These configurations are not stored in the Preferences, but are rather held in memory as Properties objects.

4.4 Choosing a configuration to use...

4.4.1 ...in i-net DesignerXML

In order to choose which configuration you wish to use in i-net DesignerXML, select the menu point “File -> i-net Crystal-Clear Options”. Here you can create, edit, delete, copy, import, and export configurations, as well as choose the configuration you wish to use in i-net DesignerXML with “Activate”.

4.4.2 ...for the Report Server (Listener)

The i-net Crystal-Clear Listener will search for its configuration in the following order. As soon as a configuration is found, it is used until the configuration is changed in any way. In this case, the configuration is reloaded as soon as a new report is requested.

1. If the Java system property “crystalclear.config” has been set, its value is used to locate a configuration. This name must have “USER” or “SYSTEM” as its prefix and then have the configuration name after a slash, for example:
“-Dcrystalclear.config=USER/myconfig”
“-Dcrystalclear.config=SYSTEM/mysystemconfig”
2. If the Java system property “crystalclear.configfile” has been set, its value is used to locate a crystalclear.properties file. This file is then imported and used as a temporary configuration called “<temp copy>”.
3. A configuration called “Default” in the User scope.
4. A configuration called “Default” in the System scope.
5. A configuration called “<Temporary Default>” in the Temporary scope.

If none of the above configurations could be found, a temporary, empty configuration is created and set. Note that this configuration will only have default configuration values and no license key, so it will not be possible to display reports with a Listener running with this temporary configuration.

To change the configuration a Listener is using once it is running, you can use the API of ConfigurationManager, e.g. ConfigurationManager.setCurrent(Configuration).

4.4.3 ...for i-net Crystal-Clear running as Servlet

The i-net Crystal-Clear servlet will search for its configuration in the following order.

1. There are two init parameters for the servlet in the web.xml. The first one has the name "crystalclear.config". This enables you to set the configuration you wish to use. Its value's format is the same as for the listener: the name must have "USER" or "SYSTEM" as its prefix and then have the configuration name after a slash, e.g.: "USER/myconfig" or "SYSTEM/mysystemconfig".

If this first init parameter was not set or did not point to a valid configuration, the second init parameter is "crystalclear.configfile" and can specify a crystalclear.properties file with its full absolute path. This file is then imported into a temporary configuration called "<imported config>".

2. The Java system parameters are checked for the same properties as for the Listener. Please consult the documentation of your application server for how to set Java system properties. (see page 7)
3. A configuration called "Default" in the User scope.
4. A configuration called "Default" in the System scope.
5. A configuration called "<Temporary Default>" in the Temporary scope.

If none of the above configurations could be found, a temporary, empty configuration is created and set. Note that this configuration will only have default values and no license key, so it will not be possible to display reports with a servlet running with a temporary configuration.

4.5 Changing Configurations “On-the-fly”

Starting with version 8 of i-net Crystal-Clear, it is no longer necessary to restart the Listener or application server after changing a configuration property - i-net Crystal-Clear automatically detects the change in the configuration and reloads the configuration the next time a report is requested.

There are two ways to change a configuration:

4.5.1 Configuration Manager Tool

The Configuration Manager can be viewed either from within i-net DesignerXML (File -> i-net Crystal-Clear Options) or by starting it with the command “java -jar CrystalClear.jar”. From within the configuration manager it is possible to create, edit, delete, and rename configurations. Remember that unless a specific configuration is manually set as default, i-net Crystal-Clear will look for a configuration called “Default” in the user or system scope.

4.5.2 API

For information on changing properties programmatically, see the API documentation to the classes `ConfigurationManager` and `Configuration`. To get started, here is a simple code sample for setting a configuration and then changing its properties:

```
ConfigurationManager manager = ConfigurationManager.getInstance();
Configuration c = manager.get(Configuration.SCOPE_USER, "myconfig");
c.put("propertyXY", "myvalue");
...
manager.setCurrent(c);
```

5 Datasources and Configuration of Driver DLL's

In version 8 of i-net Crystal-Clear, the mapping between driver dll name and JDBC driver properties in `crystalclear.properties` were completely replaced by the data sources which were introduced in version 7. A data source describes all settings for a database connection. The saving of the password is optional. Data sources are based on a similar concept to ODBC data sources.

5.1 GUI

The data sources can be edited with the Data Source Manager. This manager can be opened in i-net DesignerXML using the menu point “Database | Data Source Manager”, or with the Configuration Manager (the button on the bottom left). In the data source manager there are 3 scopes in which the data sources can be stored, just like with the configuration manager.

5.2 Old Reports and Crystal Reports Templates

Old report templates (from before version 7) and Crystal Reports templates still have the old dll alias stored in the report. In order to open and run these reports, you will need pseudo data sources. These are created by the setup or when importing old `crystalclear.properties`. It is not possible to create reports on these pseudo data sources (such as `pdssql.dll`).

The URL parameters such as “dll”, “host” and “catalog” only will work for old reports. The new parameter `datasource` works for all templates.

If this templates are saved or opened in the designer, a temporary data source is created and is stored as a copy in the report template.

5.3 API

For transferring to other systems, the data sources can be exported and imported from the data source manager. This is also possible using the public API of the class `com.inet.report.config.datasource.DataSourceConfigurationManager`.

6 Classic Viewer removed

The older Java report viewer of version 6 and older is no longer supported. If you did not migrate to the new Java report viewer in version 7, you will have to do so now.

7 JAR files for installation

The i-net Crystal-Clear runtime consists of the following libraries:

- activation.jar Activation Framework - used by scheduler
- CCLib.jar Core library of i-net Crystal-Clear
- CrystalClear.jar Report Engine - core library
- jcommon.jar Used by JFreeChart - core library
- jfreechartCC.jar JFreeChart library - core library
- jhall.jar JavaHelp - used by i-net DesignerXML
- mail.jar JavaMail - used by scheduler
- crystalclear/CC-Viewer.jar Java Report Viewer

Libraries of optional components:

- lib/commons-logging-1.1.jar used by JEuclid
- lib/Fositex.jar JDBC driver for simple text files
- lib/JBarcodeBean.jar JBarcode Java Bean
- lib/jeuclid.jar JEuclid - MathML Java Bean
- lib/mysql-connector-java-3.1.12-bin.jar JDBC driver for MySQL
- lib/Sero.jar JDBC driver for Oracle Server
- lib/Sprinta.jar JDBC driver for MS SQL Server
- lib/Syto.jar JDBC driver for Sybase Server

The following libraries are required if you like to embed the i-net Crystal-Clear report engine into a Java application or Java Servlet:

- CCLib.jar
- CrystalClear.jar
- jcommon.jar
- jfreechartCC.jar
- crystalclear/CC-Viewer.jar

If you use scheduler functionality then it is also necessary to add the files activation.jar and mail.jar to the class path.

To get these files execute the setup and install the Report Server - Runtime Library in a temporary directory.

8 New Chart API

The Chart has an entirely new API, the old one is deprecated.

The property "Compatibility Level" defines whether Charts will be converted during report loading so that they can be accessed using the new Chart API (see the class `com.inet.report.Chart2`).

If the value is set to "compliant to version 7.x" the old Chart-API (class `Chart`) can be still used for reports saved with i-net DesignerXML versions previous to version 8. You can combine also both Chart API's by using the `ChartConverter` class. See the following example code:

```
//The engine was initialized with a report with a chart in the first
//section of report header
Section section =
engine.getArea(Engine.AREA_TYPE_REPORT_HEADER).getSection(0);
//Gets the old chart element
Chart chart = (Chart)section.getReportElement(0);
//Using the old API
chart.setStyle(Chart.BAR2D_STYLE);

//Converts the old chart to new, the old chart will be replaced with
//Chart2
ChartConverter.replaceChart(chart);

//Gets the new chart element
Chart2 chart2 = (Chart2)section.getReportElement(0);
//Using the new API
BarPlot plot = (BarPlot)chart2.getPlot();
plot.setStyle(BarStyle.BAR2D_STACKED);
```

In contrast, if the value is set to "compliant to version 8.0 or later" then Charts can only be accessed using the new Chart-API (class `Chart2`).

9 Troubleshooting / FAQ

9.1 I don't see a crystalclear.properties file in our installation directory! How do I configure my i-net Crystal-Clear installation?

Starting in version 8 of i-net Crystal-Clear, crystalclear.properties are no longer used. All configurations are instead stored in the operating system's Preferences (e.g. the Registry when under Windows). To edit and manage your configurations, the easiest way is to use the Configuration Manager, by running the CrystalClear jar file (e.g. `java -jar CrystalClear.jar`).

See [Chapter 4](#) for more information.

9.2 I have entered a valid license key but still am receiving the error “No license key” or “No configuration found”!

Most likely the configuration with your license key is not being used by your i-net Crystal-Clear installation. See [Chapter 4.4](#) for information on how to determine which configuration is to be used by your i-net Crystal-Clear installation.

9.3 I don't see any database dll properties in the new configuration dialog! How do I configure my database connections?

The database dll properties were entirely replaced by data sources in version 8 of i-net Crystal-Clear. To configure your database connections, use the Data Source Manager. See [Chapter 5](#) for more information on this. Note that importing old crystalclear.properties files will cause the dll properties to be imported into data sources. See [Chapter 4.2](#) for more information on importing crystalclear.properties files.

9.4 I have an application still using the old Chart API - using these API methods now causes an error! Do I have to change all of my code to the new API?

Version 8 of i-net Crystal-Clear supports an entirely new Chart API with many new features and an improved look to the charts. It is highly recommended to switch to the new Chart API to take advantage of these new features.

If you want to, however, you can make use of the “Compatibility Level” configuration property, and/or the ChartConverter class. See [Chapter 8](#) for more information on this.