

CoSort® Version 11.0.1  
Innovative Routines International (IRI), Inc.  
January 20, 2026

NDA CONFIDENTIAL: CoSort Release Notes

This readme file contains release notes that supplement CoSort's product manual and on-line help in IRI Workbench™ for CoSort. You may print a copy of the release notes but you may not transmit, disclose or post their contents without the prior written consent of IRI, Inc. If you have any questions about the use or contents of these notes, please contact IRI:

CALL +1-321-777-8889  
EMAIL [info@iri.com](mailto:info@iri.com)  
URL <http://www.iri.com>  
WRITE Innovative Routines International (IRI), Inc.  
2194 Highway A1A, Suite 303  
Melbourne, FL 32937-4932  
United States of America

These notes describe enhancements to CoSort® for UNIX®, Linux® and Microsoft® Windows® effective since Release 10.5.2. Also note that the:

Changes from 10.01 to 10.5 are in the Version 10.5.1 readme file.  
Changes from 9.5.x to 10.0 are in the Version 10.0.1 readme file.  
Changes from 9.1.x to 9.5.x are in the Version 9.5.3 readme file.  
Changes from 8.2.x to 9.1.x are in the Version 9.1.3 readme file.  
Changes from 8.1.x to 8.2.x are in the Version 8.2.3 readme file.  
Changes from 7.5.x to 8.1.x are in the Version 8.1.3 readme file.  
Changes prior to 7.5.x may also be available from [cosort@iri.com](mailto:cosort@iri.com).

Most of these files are available upon request from CoSort users who are upgrading from older versions of CoSort.

CoSort Version 11 is currently supported on Windows and Linux x86 hardware, and can be made available for AIX, Solaris, HP-UX and z/Linux. For IBM® iSeries® platforms running Linux or OS/400® PASE, refer to the Unix instructions in this file and other CoSort documentation. See the list of historically-supported CoSort platforms under [www.iri.com/products/cosort/platforms-pricing](http://www.iri.com/products/cosort/platforms-pricing).

The remainder of these release notes are divided into the following sections:

- (1) Product Directory Organization
- (2) Setup Program and Licensing Procedure
- (3) Multi-threaded Performance and Tuning
- (4) Product Enhancements and Corrections
- (5) Known Issues and Workarounds
- (6) Copyright and Trademark Attribution

## (1) PRODUCT DIRECTORY ORGANIZATION

CoSort software will be installed into the default directories shown below. Both Unix and Windows users can utilize the same job scripts and performance parameters across both platforms.

The base directory where CoSort is installed must be set as the value of an environment variable named `COSORT_HOME`. This location is referenced by the environment variable `%COSORT_HOME%` on Windows, and `$COSORT_HOME` on other platforms.

On Unix, the system administrator typically installs CoSort into a base directory named `cosort110/` below any path, such as `/usr/local/cosort110`. This location is then set as the value of `$COSORT_HOME`. On Windows, the default installation behavior is to create a base directory at `C:\IRI\cosort110`. This location is then set as the value of `%COSORT_HOME%`.

Note that your operating system may not provide sufficient privileges for the CoSort installer/user to update the `C:\IRI\etc\license` file as part of the default `cs_setup` program run from the command line or IRI Workbench. See your installation instructions if you need to manually apply the license key string in your `etc/cosort.lic` file.

The following list of subdirectories are relative to the base install directory, which is the value of `COSORT_HOME`:

Directory	Contents
-----	-----
<code>bin</code>	executables, conversion tools
<code>docs</code>	manuals, readme file, license agreement
<code>etc</code>	tuning, license, error and log files
<code>examples</code>	paths to sample specifications
<code>include</code>	headers, defines, errors, etc.
<code>lib</code>	user exit, plug-ins, and API libraries
<code>lib/modules</code>	external libraries that auto-load at runtime
<code>sets</code>	sample files for lookup matching, pseudonymization or synthesis

IRI Workbench, Built on Eclipse™, is the Integrated Development Environment (IDE) for the CoSort Sort Control Language (SortCL) program. Compatible with Windows and Linux, its primary purpose is to create, import, modify, save, print, share, and launch SortCL job scripts. Jobs can be executed locally, or on a remote host over a TCP/IP connection. IRI Workbench is available at no cost to CoSort users. IRI Workbench can be installed separately, or as part of the CoSort full install package for Windows. The default install location for IRI Workbench will vary by platform, and whether it is installed for all users, or a specific user.

## (2) SETUP PROGRAM AND LICENSING PROCEDURE

On both Unix and Windows systems, CoSort uses similar installation and configuration programs to combine loading, licensing, and resource tuning. The setup program runs automatically at first time installation, and can be re-run when updating the license or tuning parameters. The `cs_setup` program will prompt you for responses based on your current configuration and job requirements; see (3) below.

First time installers must obtain a 3-part license (activation) key from IRI to run CoSort programs or enable API calls to CoSort libraries. The key allows you to easily conform to the terms of an evaluation or permanent license, and allow upgrades to CoSort or your hardware, usually without new shipments.

CoSort 11 for Unix and Windows uses a central license file named "cosort.lic" which contains a license key specific to each machine, or in the case of expiring floating licenses, central to a container. By keeping licensing information central to the machine instead of the executable, users can update specific files via downloaded patch files, without the need to reinstall, re-register, or re-key the product. Prior to Version 10, license and tuning information was stored in the registry on Windows.

### (3) MULTI-THREADED PERFORMANCE AND TUNING

To scale high-volume data processing requirements for very large data sets and data warehouses, CoSort performs sorting and related data transformations across multiple cores on systems where supported. Since Version 10, CoSort has improved automatic management of system resources to maximize the efficiency of high-volume sorting jobs along with other applications running in a multi-threaded, multi-user environment.

Both the Unix and Windows versions of CoSort help set global default system tuning values interactively during initial setup. These values are stored in a CoSort Resource Control file named "cosort.rc" in \$COSORT\_HOME/etc. Prior to Version 10, the default tuning file on Unix and Linux platforms was named "cosortrc". This file name is still supported on all platforms in Version 11, but "cosort.rc" is now preferred.

Global tuning values can be overridden by creating a local tuning override file. On Unix, it is a .cosortrc file in the working directory. On Windows, it's a cosort.rc file in the same directory as a CoSort executable on Windows. See Appendix D in the CoSort manual for resource control setting locations and parameter search priorities.

The following resource control settings are commonly used in CoSort

Control Name <value>	Default Value	Description
-----	-----	-----
THREAD_MAX <count>	As licensed	max # of sorting CPUs
THREAD_MIN <count>	1	min # of sorting CPUs
MEMORY_MAX <AUTO,MINIMIZE,#,%>	AUTO	RAM for sort buffers
WORK_AREA <directories>	./	overflow (temp) path/s
BLOCKSIZE <bytes>	1200/3584KB	size of I/O buffers
MINIMUM_YEAR <2-digit year>	70	sliding century window
MONITOR_LEVEL <0-9>	1	running message detail
ON_EMPTY_INPUT <option>	PROCESS_WITH_ZEROS	sortcl output displays
ON_WORKAREAS_FULL <option>	ABORT	pause/resume behavior
OUTPUT_TERMINATOR <option>	INFILE	VL record terminators

During setup, allow the MEMORY\_MAX value to be set to AUTO. If however you will be running multiple, large sortcl jobs at the same time, set MEMORY\_MAX to MINIMIZE to automatically reduce the use of RAM for more efficient operation in that environment.

Additionally available, advanced RC settings documented in the manual include:

ON\_MISSING\_OUTLENGTH

AIO  
AIO\_BUFFERS  
AIO\_RETRY\_TIMEOUT  
LEGACY\_NUMERIC\_OUTPUT  
ON\_EMPTY\_OUTPUT  
SUMMARY\_OVERFLOW\_BREAK  
ENDIANNESS  
ON\_COLLATION\_FAILURE  
ON\_CONVERSION\_FAILURE  
PREVENT\_DOUBLE\_TERM  
USE\_RECORDCOUNT\_API  
COMPRESS\_WORKFILES  
ON\_EMPTY\_OBJECT\_VALUE  
PRESERVE\_ERROR\_LOG  
ZIP\_OUT\_LEVEL  
ON\_FIELD\_OVERFLOW

The following RC settings are deprecated; i.e., no longer supported:

AUDIT is replaced by the JSON log defined in the OGS Policy File  
LOG is replaced by the JSON log defined in the OGS Policy File  
MEMORY\_PERPROCESSOR\_MIN  
MEMORY\_PERPROCESSOR\_MAX  
EXTERNAL\_MAXMEM  
MERGE\_WORKFILES\_PERTHREAD\_MIN  
MERGE\_WORKFILES\_PERPROCESSOR\_MIN  
BUFFER\_IPC\_MIN  
BUFFER\_IPC\_MAX  
BUFFER\_IO  
OUTPUT\_PROCESS  
AUTO\_MEM  
PERCENT\_SHARED  
CHECK\_DISK\_FREE  
IN\_ACCEL  
OUT\_ACCEL  
OUTPUT\_PROCESS  
PROCESSOR\_MIN is replaced by THREAD\_MIN  
PROCESSOR\_MAX is replaced by THREAD\_MAX  
CONSERVE\_CPU is replaced by COMPRESS\_WORKFILES.

Additional performance recommendations:

WORK\_AREA  
-----

For the best sort performance on multi-CPU systems, specify up to 4 directories on different available physical disks. For example, to specify two disks in an RC file:

```
WORK_AREA      /export/home1/sorttemp
WORK_AREA      /export/home2/sorttemp
```

Sort work files will be read simultaneously from both. Also important, try not to specify work areas that are on the same physical disks as the input or output files.

## LICENSE LIMITS

-----

The values in your resource control files only apply if your system, job, and CoSort license allows them. For example, you cannot specify `THREAD_MAX 8` or use more than 4 hyperthreaded CPU cores if your company has only acquired a license to use 4 threads.

## GET HELP

-----

Manual system tuning can override automatic and default settings provided by IRI for your environment. Only qualified users and system administrators should modify the values in resource control files, since changes can impact sort and non-sort jobs. Preceding configuration file names with a dot can hide them from normal directory listings. Either your IRI representative or engineer (email [support@iri.com](mailto:support@iri.com)) can review your resource settings and system information, and recommend changes. Also, Appendix D in the CoSort User Manual & Programmer's Guide explains how to set and override RC values, the order of precedence for multiple resource settings, and additionally available global resource or SortCL application specific parameters. On all platforms, see settings in effect prior to execution by entering "sortcl /rc".

## (4) PRODUCT ENHANCEMENTS (since 10.5.2)

### a) Operational Governance System (OGS)

Introduced with CoSort 11 is a new runtime CLI framework for enforcing granular RBAC policies and logging SortCL-compatible jobs in production. The system defines user, group and role permissions for specific scripts and script elements, generates a comprehensive new JSON performance and audit log for each run, can detect job script tampering through hashed-based signatures, manages date-based log retention behavior, supports rudimentary data descriptions (cataloging), and provides menu-driven policy file management and log wrangling tools to facilitate reporting. See separate OGS documentation (not the CoSort manual) for details.

### b) Data Class Attribution in /FIELD Statements

SortCL jobs now recognize optional `CLASS=DataClassName` declarations in /FIELD statements to support assignments between Workbench-defined data class names and specific source and target field names. IRI Workbench currently produces this in FieldShield job scripts for deterministic data masking jobs, but the automatic assignments can be extended to other wizards. This feature supports role-based access controls to data classes to more centrally assign user, group, and/or role-related read/write (action) permission to PII instead of less convenient field-specific permission definitions. Additional IRI Workbench job wizards will add CLASS to additional SortCL-based job scripts for related purposes in the subsequent releases.

### c) New SortCL Runtime Log Format

A comprehensive, per-job JSON log for performance analysis and metadata (audit) is now produced by default, and written into the `cosort110/logs` folder. These logs can be customized as to content in OGS Policy Files. The JSON log contains all of the data in, and obsolesces, all prior CoSort log formats (including /AUDIT XML, `cosort.log`, and SortCL /STATISTICS files). With the exception of `.cserrlog`, the old logs are deprecated; i.e., still available (but not supported) as of CoSort 11.

### d) Cloud Store Support

Read and write support in Windows and Linux for flat files in AWS S3, GCP, Azure Blob, and OneDrive sync. For more information, see:

<https://www.iri.com/blog/data-transformation2/voracity-software-support-for-cloud-file-stores/>

e) Nested Functions

Nested functions in the same /FIELD statement now support the combined application of more than one data transformation function at a time. For example, you can format the encrypted result of a substring value in one /OUTFILE field declaration.

f) Null Handling

Improvements were made to processing null values in both files and database tables.

g) ODBC2DDF Enhancement

A new -M argument flag is available in the CLI version of the utility to multiply a numeric value in a database by the number of places needed to remove the decimal but retain the digits. This enables conversion of the value into a packed decimal (MF\_CMP3) data TYPE in SortCL outfile /FIELD statements.

h) RDB Catalog Support

SortCL and Workbench now support /INFILE & /OUTFILE designations of RDB sources and targets for Unity Catalog environments in PostgreSQL, Databricks, DB2, etc. via new Catalog.Schema.Table syntax. Non-catalog environments are still supported.

i) AI JCL Sort Parm Migration (Underway)

A new JCL sort parm conversion portal page at the IRI Web Site is in development to exceed the capabilities of the older CLI MV2SCL utility that produces functionally equivalent CoSort SortCL scripts given rudimentary z/OS DFSORT parameters. Test here: <https://www.iri.com/translator/?code=3d5ec87b65638f363e9cad825cb1d36dbce6595c>

j) New SET Files

Set files are used primarily in data classification (discovery), pseudonymization (replacement) and test data generation (synthesis) jobs powered by CoSort-compatible jobs scripts. CoSort 11 and IRI Workbench distributions contain new set files for multiple countries and global standards as well as a greatly enhanced corpus of real-names based on US government databases. See the sets subdirectory for contents and this article for more information on set files generally: <https://www.iri.com/blog/test-data/all-about-iri-set-files-a-primer/>

k) Type Mapping Wizard (for RDB Data) in Workbench

Using CoSort to integrate and transform data between relational database (DB) platforms can be complicated when those DBs do not use the same data types. A DB Data Type Mapping wizard is now available in IRI Workbench to help solve this problem for SortCL-driven data management jobs with different sources and targets. <https://www.iri.com/blog/data-transformation2/mapping-db-data-types/>

l) New Data Classification Infrastructure

IRI Workbench now supports both project-specific or multi-project data class matchers and data rules for masking, migration, and synthesis in IRI Workbench that ultimately manifest in consistent application of defined functions at the field level in one or more SortCL-compatible job scripts. This obsolesces data class rules that were originally defined in the Workbench Preferences menu. For more information, see: <https://www.iri.com/blog/data-protection/iri-data-classification/>.

m) IRI Workbench supporting CoSort 11 and other IRI tools is now based on Eclipse 2025-11 and 64-bit Java 21.

## (5) KNOWN ISSUES AND WORKAROUNDS

The following limitations apply to the build accompanying this file. Therefore, the information below is subject to change without notice. Please contact your IRI representative or support@iri.com for updates.

### a) 64KB record length limitation

All SortCL process types currently support records up to 65,535 bytes. In cases where the length of the input field exceeds this limit, the file can still be processed with a specially compiled version (available upon request), and sort keys must be in the first 64KB of each record. In the case of JSON, those larger files can also be processed if the SortCL script selects a subset of value paths that does not exceed the maximum input record length.

### b) Unsupported BLOB and CLOB Columns

These columns are unsupported for database and file based operations in SortCL but can be masked in IRI DarkShield. IRI is developing a SortCL bypass for database column objects that will point to binary data in stored standalone files.

### c) Order of expression

When cross-calculations involving functions are specified in SortCL /FIELD statements in an order other than as documented, a syntax error will be issued rather than allowing the expression to produce an incorrect result. Further development may support alternate orders.

### d) Mixed Record Formats in Multi-Table Joins

When you have different record formats (i.e., fixed vs. floating) in joins involving three or more files, sortcl completes the join only when the last-named join file is the only input source with a different record format. Workaround: Pre-process your differently formatted input sources so they are all in the same format prior to performing a join on three or more inputs.

### e) Unsupported Options in Join Scripts

/INREC is not supported. Workaround: Specify input files needed only for join conditions and output display purposes. /HEADREAD and /HEADWRITE and /TAILREAD and /TAILWRITE cannot be used in a join because they are currently being processed. Workaround: Remove header and footer records in prior processing steps and reapply them if needed in a subsequent job script.

### f) Overdefined Fields in /INREC

The derived\_name=field\_name convention is not supported in the INREC phase when the derived name is referenced in the output. Note that this does not affect the use of a derived\_name=field\_function or expression. Workaround: Since the feature is typically used to over-define the same field, redefine the field with the derived\_name in the /INFILE section. A derived field name should not be given the same name as a source field name so as to prevent a circular reference error.

### g) Record Loss in ODBC /CREATE Operations

Rows read from and written to the same table may be lost when /CREATE and /REPORT are specified in a SortCL job. Changing either the job action to /SORT (which can change the order of the rows), or the write statement to /UPDATE will resolve this. Note also that /APPEND is the default behavior for /PROCESS=ODBC operations, while /CREATE is the default for files (where the above issue/workaround does not apply).

#### h) Formatting of XLS and XLSX cell data

Special formatting of numeric cells, such as placing a dollar sign before the numeric value, is not preserved. The actual raw cell value can differ from what is shown on the screen in Excel after formatting. For example, a currency cell with a value of 5 displays as \$5.00, but the actual cell value is 5. Only the raw cell values are read from XLS and XLSX. As an alternative, the cell format can be changed to text so the raw cell value is guaranteed to match what is shown in the spreadsheet view. For dates, times, and timestamps (which are also stored as numeric values internally in Excel), there is another alternative. If a corresponding SortCL date, time, or timestamp datatype is specified for the input field then the Excel date/time serial number will be converted to the specified SortCL date, time, or timestamp datatype.

### (6) COPYRIGHT AND TRADEMARK ATTRIBUTION

Copyright © 1978,2026 Innovative Routines International (IRI), Inc.  
All rights reserved.

- CoSort® and Voracity® are registered trademarks, and IRI Workbench™ is a trademark, of IRI.
- Eclipse™ and Built on Eclipse™ are trademarks of the Eclipse Foundation.
- IBM®, iSeries® and OS/400® are registered trademarks of the International Business Machines Corporation in the U.S. and other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Microsoft®, Windows®, and Excel® are registered trademarks of Microsoft Corporation in the U.S. and other countries.
- UNIX® is a registered trademark of The Open Group.