CoSort® Version 10.5.1
Innovative Routines International (IRI), Inc.
June 30, 2021

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
        URL     http://www.iri.com
        WRITE   Innovative Routines International (IRI), Inc.
                2194 Highway A1A, Suite 303
                Melbourne, FL 32937-4932
                United States of America
```

The release notes are divided into these sections:

These notes describe enhancements to CoSort® for UNIX®, Linux® and Microsoft® Windows® effective since Release 10.0.1. Also note that the:

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.

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

For IBM® iSeries® users running Linux or OS/400® PASE, refer to the Unix instructions in this file and other CoSort documentation. See the list of currently-supported CoSort platforms under www.iri.com/products/cosort.

(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 cosort105/ below any path, such as /usr/local/cosort105. 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\cosort105. 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
bin	executables, conversion tools
docs	manuals, readme file, license agreement
etc	tuning, license, error and log files
examples	paths to sample specifications
include	headers, defines, errors, etc.
lib	user exit, plug-ins, and API libraries
lib/modules	external libraries that auto-load at runtime
sets	sample files for masking and test data jobs

IRI Workbench, Built on Eclipse™, is the Integrated Development Environment (IDE) for the CoSort Sort Control Language (SortCL) program. Its purpose is to create, import, modify, save, print, 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 10 for Unix and Windows uses a central license file named "cosort.lic" which contains a license key specific to each machine. 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. CoSort since version 10 has improved automatic management of system resources to maximize the efficiency of high-volume sorting jobs along with other applications running in a multi-tasking 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 10, 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 10, for both Unix and Windows platforms:

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
LOG [path]<file name>	[no entry]	continuous log file
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
AUDIT [path]<file name>	[no entry]	XML audit file

As of CoSort 10.5, the following additional settings are also recognized:

Control Name <value>	Default Value	Description
LEGACY_NUMERIC_OUTPUT <switch>	0	When set to 0 or OFF

(0 is the default) NUMERIC output is reformatted to assert output field precision. When set to 1 or ON, NUMERIC output precision is only enforced when the field precision or data type has changed.

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
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 no longer supported:

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 deprecated and has been replaced by THREAD_MIN
PROCESSOR_MAX is deprecated and has been replaced by THREAD_MAX
CONSERVE_CPU has been 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) can review your resource settings and system information, and recommend changes.

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 10.0.1)

a) Database Lookups
As an alternative to drawing values from external set files, it is now also possible in SortCL /FIELD statements to use some existing SET file attributes in conjunction with values in a DSN-connected database table.

b) Data Vault Support
IRI Workbench now provides a wizard in the Voracity menu to generate a Data Vault table with RDB production schema or synthetic test data.

c) Sybase and MS SQL Date Handling
Current time, data and timestamp data types are now supported.

d) EPWD, EUID, EPASS, and cspasswd
Encrypted usernames and passwords may be specified in the ODBC connection string by using "EUID:encrypted_username" and "EPWD:encrypted_password" in place of UID and PWD followed by clear text.

Encrypted passphrases may be specified as an argument to encryption functions with the syntax "EPASS:encrypted_passphrase" in place of the clear text passphrase.

The "cspasswd" executable is included in the CoSort product distribution to be used to encrypt the password or username for EPWD, EUID, and EPASS.

e) Encrypted credentials in ODBC connection strings

User IDs and passwords for database connections can be optionally encrypted when used in a job script.

f) Empty tags or attributes in XML output
The existing resource control setting, ON_EMPTY_OBJECT_VALUE, may be set to REMOVE_EMPTY to prevent the writing of tags and attributes with empty values in XML output. This setting was previously only applicable to JSON formatted output.

g) Legacy Numeric Output
The new resource control setting, LEGACY_NUMERIC_OUTPUT, is available to revert the reformatting of output records to the way they were handled in CoSort 9. When set to 0 or OFF (the default) NUMERIC output is reformatted to assert output field precision. When set to 1 or ON, NUMERIC output precision is only enforced when the field precision or data type has changed.

h) /PROCESS=ASN.1
SortCL now supports the processing of files based on any ASN.1 schema and encoding. asn1_2ddf, a new metadata utility for ASN.1 encoded files, has been developed to aid in gathering metadata from such files for use with SortCL.

i) /PROCESS=XLS and /PROCESS=XLSX
Reading and writing cell data from .xls and .xlsx spreadsheet files is now supported. Additionally, xls2ddf, a new metadata utility for XLS and XLSX files, is included.

j) Location DEF Expansion
SDEF has been added as a new field location definition to be used with XLS and XLSX formats.

CDEF has been added as a new field attribute to support CSV output column names that do not comply with field name restrictions. This includes UTF-8 characters and names with leading digits or embedded white space.

k) Eclipse Upgrade
IRI Workbench supporting CoSort 10 and other IRI tools is now based on Eclipse 2020-06 and 64-bit JRE 1.8.

l) New Integrations with Splunk and Value Labs
A purpose-built app for Splunk directly executes and consumes job data from SortCL and compatible IRI programs. The ValueLabs Test Data Hub can also create, modify, and launch SortCL scripts for data masking and synthesis jobs.

m) Discovery Dashboards in IRI Workbench
A number of visualizations can now be generated from the Data Class Schema (and Directory) Search wizards in addition to their traditional log output.

(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) DDF for JSON files and MongoDB collections
A standalone command-line utility for automatically generating SortCL data definition

files does not accompany the initial release of CoSort 10. A DDF is instead generated in the IRI Workbench metadata discovery dialog. Note that JSON output is formatted using the JSON path specified in the output section of the script. When creating or modifying the JSON output format, be sure to specify JSON paths in their valid sequence. That is, there can be no duplicate value paths or paths to a non-flat value. If fields in the output section are not in a valid sequence, invalid JSON output format will be produced.

b) 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.

c) Unsupported BLOB and CLOB Columns
These columns are unsupported for database and file based operations.

d) 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.

e) 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.

f) 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.

g) 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.

h) Record Loss in ODBC /CREATE Operations
Rows read from and written to the same table will 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).

i) 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

- 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.