
extend[®]7

Interoperability Series

ACUCOBOL-GT users now have two ways to dramatically speed Vision index and flat-file data processing for legacy migrations, data warehouse (ETL), BI reporting, and testing.

CoSort 'SortCL' Programs

Speed and combine Vision and flat-file sorts, merges, joins, aggregations, reports and loads with external Sort Control Language (*SortCL*) jobs. Collate, convert, and create Vision files from other formats (e.g., ISAM).

High-Volume Sort Engine Alternative

Compile and link CoSort's high performance sort engine directly into ACUCOBOL-GT. With no changes to your source programs, you can dramatically accelerate sort runtimes and sort-dependent operations downstream.

**CoSort[®] Delivers Faster
Sorts and Vision File Handling
For ACUCOBOL-GT[®] Users**



CoSort: It's About Time

- **CoSort is the fastest parallel sort engine for UNIX**
Customer benchmarks prove it: On 6-CPU Sun Fire 15K: 2.4GB in 39 seconds. On a 4-CPU IBM p690: 1GB in 12 seconds. On a 2-CPU Sun V440: 1GB in 26 seconds vs. 62 for the competitor.
- **CoSort combines sorting with transforms and reports**
CoSort's Sort Control Language ([SortCL](#)) program can also simultaneously join (match), extract, clean, reformat, merge, aggregate, cross-calculate, type-translate, and report – all in one I/O pass.
- **CoSort supports all data types and sizes**
CoSort can collate, convert, and display more than 100 data types and record formats, including EBCDIC, IP addresses, CSV, UNICODE, COBOL, VARCHAR, and timestamps.
- **CoSort speeds database operations**
CoSort can 'select', 'join', 'order by', and 'group by' faster than an RDBMS. De-duping, sorting and aggregating millions of rows also speeds bulk loads, reducing DB index overhead and storage space.
- **CoSort is easy to learn and use**
CoSort's *SortCL* is an explicit, high-level data definition and manipulation language familiar to mainframe sort and SQL users alike. *SortCL* is widely used for metadata repositories, data warehouse staging, legacy sort migrations, and reporting. It is leveraged in [Fast extraCT](#) and [RowGen](#), and ETL tool users can automatically create *SortCL* data definition files through [MIMB](#).
- **CoSort offers multiple UIs and APIs**
CoSort's standalone utilities and program calls accommodate all levels of user expertise and job complexity. You can also use CoSort's drop-in sort replacements (and conversion tools for legacy sort parms and metadata) to seamlessly upgrade from slower third-party sort functions.

Fast, Combinable Data Transforms in the File System

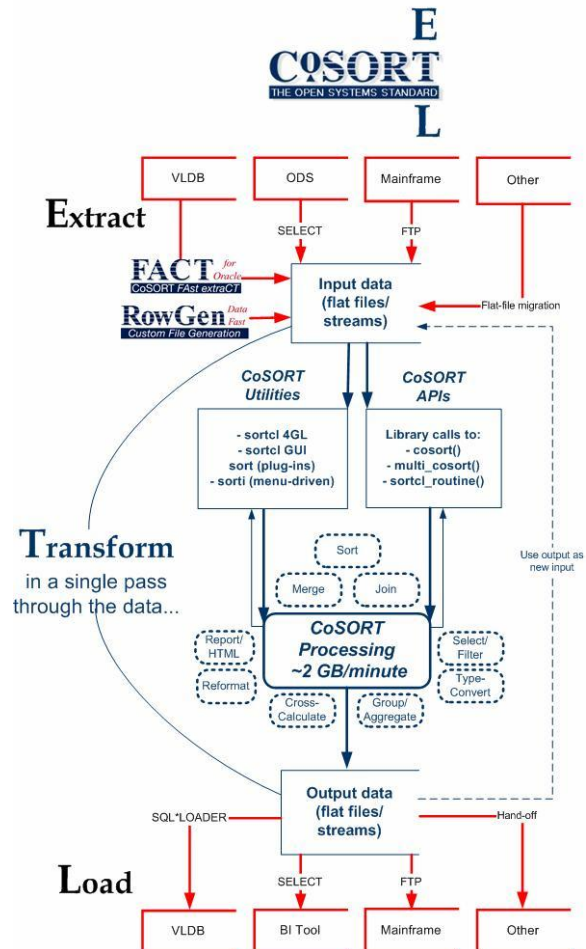
Calling CoSort's [SortCL](#) program in COBOL or shell scripts can speed [ETL](#) jobs. Plus, with CoSort's [cob2ddf](#) tool or Meta Integration's new [export bridges](#) (for third-party metadata) to [SortCL](#)'s data definition file (DDF) format, you can re-use your existing field layouts!

[SortCL](#) users can usually combine the tasks below in the same job script and I/O pass:

- **Extracting (Selecting)** – [SortCL](#) accepts any number and size of Vision or sequential files, pipes, DB feeds (e.g. Oracle via IRI's [FAst extraCT](#)), and/or custom input routines.
- **Cleansing (Filtering)** – [SortCL](#) conditionally includes or omits records through logical evaluation, and can filter by bytes, records, duplicates status and other criteria. Duplicate records can be discarded or saved. Such data reduction speeds downstream [SortCL](#) jobs:
- **Sorting (Permuting)** – [SortCL](#) employs CoSort's parallel-processing, coroutine sort engine to rapidly reorder millions of records on specific keys and sequences. Results go to one or more formatted outputs, including: roll-up reports, joins, loads, BI tool hand-offs, etc.
- **Matching (Joining)** – [SortCL](#) can match two sorted files at I/O speed, using SQL join syntax and your condition logic. A [SortCL](#) join has run up to 20X faster than an RDBMS join.
- **Aggregating (Grouping)** – [SortCL](#) can count, summarize, average, and find the maximum and minimum values based on inter- and intra-record break conditions to drill-down multiple classes of field data. You can also display accumulating aggregates, and even aggregate on cross-calculated values.
- **Remapping (Reformatting)** – [SortCL](#) can remap, resize, relocate, and convert data types. Reformat entire file process types (e.g., MF-ISAM to Vision, Line Sequential to CSV). Rewrite or strip header and footer records; find, replace, and process sub-strings; insert special formatting characters and environment variables (including HTML statements for web reports); find and replace values, and, cross-calculate between field and constant values.

- **Type Converting (Translating)** – [SortCL](#) can also convert field data from one type to another, such as EBCDIC to ASCII, or mixed packed decimal to signed and zoned decimal, ISO to Japanese timestamp, and so on. Many mainframe binary forms are undesirable for data propagation, mining and access tools on open systems -- which is also why [SortCL](#) processes and creates CSV-formatted records as well.

- **Loading/Indexing (Piping/Sequencing)** -- [SortCL](#) can sequence outbound records and pipe them directly to a loading operation. In some databases -- CA-IDMS, Cincom Supra and IBM DB2 -- CoSort has been called directly to speed table or index population. In Oracle, Sybase, Red Brick and MS SQL Server environments, pre-sorted loads are up to 90% faster than unsorted bulk loads.



CoSort Collates, Converts and Creates Vision Files

In addition to Vision file sorting, CoSort's SortCL can convert data in other file formats into Vision files and vice versa, letting you move between index and sequential processing with ease.

A CoSort Sort Control Language program can define the layouts and transformations of one or more Vision file sources or targets. In this *SortCL* job script example, a Vision file is being sorted on two keys and converted to a standard record sequential (flat) file. Note also that the pipe-delimited input (as represented after a *vutil* analysis of the source file) is also being converted to fixed-position, fixed-length records on output.

Reversing the */PROCESS* (i.e. specifying Vision on output, and nothing or Record on input) would create a Vision index file from a sequential file. After data is converted to Vision format, use *vutil* to display its details.

```
/INFILE=chiefs_sep
/PROCESS=VISION
/FIELD=(lname, POSITION=1, SEPARATOR='|')
/FIELD=(fname, POS=2, SEP='|')
/FIELD=(years, POS=3, SEP='|')
/FIELD=(party, POS=4, SEP='|')

/SORT
/KEY=lname
/KEY=years

/OUTFILE=chiefs_sep.out
/PROCESS=RECORD
/LENGTH=44
/FIELD=(lname, POSITION=1, SIZE=20)
/FIELD=(fname, POS=22, SIZE=10)
/FIELD=(years, POS=33, SIZE=4.2, NUMERIC)
/FIELD=(party, POS=40, SIZE=3)
```

And, CoSort is a High-Performance Sort Alternative

Simply compile and link CoSort's acuCoSort.a into ACUCOBOL-GT for dramatic improvements in sort runtimes and sort-dependent jobs, without making any changes to your existing programs.



In extend Version 6.2 - 7.2, third-party sorts can also be called from ACUCOBOL-GT. This feature allows CoSort's optimized sorting engine to be used in place of the native ACUCOBOL-GT sort verb on UNIX.

Fixed-key, ASCII Sorting on 2-CPU Sun 280R

Input File Size > Target:	500 MB	1 GB	4.65 GB
ACUCOBOL-GT sort	11m30s	23m	300m
CoSort acucosort.a	4m	8m	40m
CoSort SortCL	2m	4m	19m

CoSort's sort performance on UNIX is second to none. Its flexible licensing and tuning features allow you to exploit whatever CPU, RAM and disk resources you desire for CoSort operations in a multi-user mix.

COSORT

THE OPEN SYSTEMS STANDARD

TECHNICAL SPECIFICATIONS

Supported Platforms:

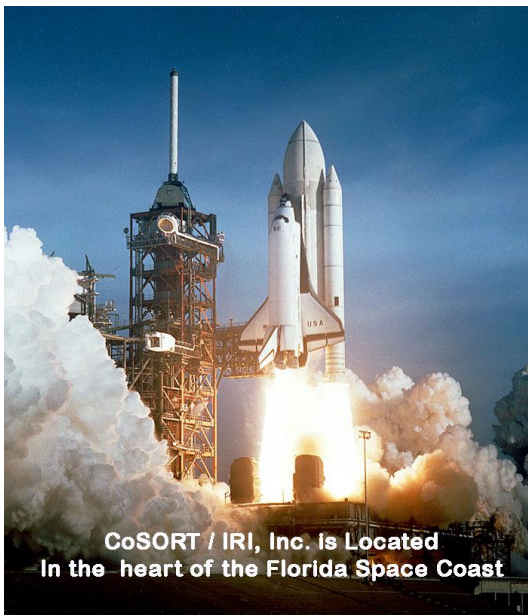
UNIX (AIX, HP-UX, Solaris, Tru64 & more);
all Linux flavors on x86, Itanium, IBM x/p/i/z;
FreeBSD; Windows (XP, NT, 2000/3 Server)

Supported Functions:

- Sort, Merge, Select, Check, Join, Type-Convert, Cross-Calculate, Aggregate (Sum), Reformat, Sequence, Report
- MVS/VSE JCL sort parm conversion
- Third-party sort tool replacements
- Third-party file metadata conversion

Data Sources & Targets:

- DBMS: Oracle 7.1-11 – via [FACT](#)
- Sequential (Flat) Files
- Micro Focus Var. Length & ISAM Files
- ACUCOBOL-GT Vision Index Files
- VSAM (for Clarity MBM)
- Unisys Variable Blocked Format
- Microsoft Comma-Separated Values
- W3C Common & Extended Log (Web)
- ACUCOBOL-GT, MF and Liant COBOL
- US, Europe, ISO, Japan Timestamps
- IP Addresses, Whole Number



For more details on CoSort, as well as [Fast extraCT](#) for Oracle and [RowGen](#) test data and file synthesis, please call us:

INNOVATIVE ROUTINES INTERNATIONAL (IRI), INC.
2194 Highway A1A
Suite 303, Atlantis Center
Melbourne, FL 32937 USA

The new face of COBOL®

ACUCORP®

Complementary Technology Partner

TEL 321-777-8889
USA 800-333-SORT
FAX 321-777-8886

Email info@cosort.com
URL www.cosort.com



Agents license and support IRI tools in:

- Argentina
- Australia
- Brazil
- Canada
- Chile
- China
- Finland
- France
- Germany
- Greece
- India
- Indonesia
- Israel
- Italy
- Japan
- Korea
- Latvia
- Malaysia
- Mexico
- Netherlands
- New Zealand
- Philippines
- Portugal
- Singapore
- Spain
- South Africa
- Switzerland
- Taiwan
- Thailand
- United Kingdom

© 2007 INNOVATIVE ROUTINES INTERNATIONAL (IRI), INC.
All rights reserved. CoSort is a registered trademark of IRI. Acucorp, ACUCOBOL-GT and extend are registered trademarks of Acucorp, Inc. All other company, brand or product names may be trademarks or registered trademarks of their respective holders.