

Changing Your Sort Engine Without Changing Your Batch

A buyer's guide to the technical and contract work of replacing a commercial sort product, on the mainframe and off it

Bill Ahlbrandt

President and CEO, Ahlbrandt Software

Abstract

Most shops that run IBM mainframe batch depend on a commercial sort product, and few have changed it in decades. When a renewal, a platform migration or a consolidation program brings the question up, people usually ask about syntax: will the new engine accept the old control statements? This paper argues that syntax is the smaller part of the problem. It maps the ways batch reaches the sort, shows where the dialects differ, and explains why return codes, messages, output order and installation defaults make up an interface your schedulers depend on. Then it gives a testing method and a contract checklist, and ends with a phased plan. The main finding: you can almost always put a replacement behind the existing program name, so JCL and programs don't change. It's only safe to do that once you've taken a census of what your batch actually uses, and agreed terms that let both products run side by side while you prove they match.

What I found

- Batch reaches sort by **more routes than the JCL shows**: EXEC PGM=SORT, tool front ends like ICETOOL, COBOL and PL/I sort statements with their exits, assembler calls and other vendors' utilities. A replacement has to answer on every route you use, or you have to change the route.
- DFSORT, Syncsort MFX and CA Sort share a **large common core, and they still aren't identical**. The risk is in extended reformatting functions, tool front ends, user exits and the installation defaults that never show up in a job. Syncsort's distributed syntax is a different language altogether.
- The program name is only half the interface. Schedulers, automation and downstream jobs read the **return codes, message IDs and the order of equal-keyed records**. One empty input file can end with return code 0, 4 or 16 depending on a single installation option.
- You have to **prove it on your own statements and data**, at production volume and through the failure paths, with a comparison that allows for equal keys legitimately coming out in a different order.
- **The contract dates usually set the timetable**. Settle notice periods, the right to run both products in parallel, audit obligations and support terms on both contracts before the technical work starts.

1. Who sells sort

Sort was one of the first pieces of system software sold separately from the machine. IBM shipped its sort/merge program with the operating system until it moved to chargeable products,¹ and Computer Associates got its start in 1971 selling CA-SORT in Europe as a plug-in replacement for the IBM sort.² So the idea of swapping one sort for another behind the same name is as old as the independent software business. I've been working on mainframe sort for more than thirty years, and it was an old idea when I started. Five decades on, the market falls into four groups.

- **IBM DFSORT.** In IBM's words, "DFSORT is an optional feature of z/OS",³ one of the optional priced features of z/OS, program number 5650-ZOS.⁴ It includes the ICETOOL and ICEGENER utilities.³ Shops that standardize on another product usually keep DFSORT around anyway, because Db2 utilities call it through DFSORT-only aliases, licensed or not, according to IBM.⁵
- **Precisely Syncsort MFX.** Syncsort renamed itself Precisely in May 2020 after buying the Pitney Bowes software and data business.⁶ Its z/OS product, Syncsort MFX, is positioned as a sort, copy and join product with optional zIIP offload.⁷ The distributed line started out as Syncsort for UNIX and Windows, became DMExpress, and is now Connect ETL.⁸
- **Broadcom CA Sort.** CA Technologies became part of Broadcom in November 2018.² Broadcom's documentation for CA Sort sits in its legacy bookshelf collection,⁹ so check the current release and support status with Broadcom directly.
- **Runtimes and independent engines off the mainframe.** Rehosting environments come with their own sort. Micro Focus Enterprise Server, owned by Rocket Software since April 2024,¹⁰ supplies MFJSORT, which answers to SORT and DFSORT, and MFSORT.¹¹ AWS Mainframe Modernization provides utilities that answer to SORT, SYNC SORT and ICEMAN.¹² Independent engines either run mainframe statements directly or translate them, the way IRI does for CoSort.¹³ The open-source GnuCOBOL project's GCSORT still lists DFSORT functions like PARSE and FINDREP as feature requests.¹⁴

Why shops think about switching

Four reasons keep coming up. The first is the **renewal**, which makes people ask what the product is actually used for. The second is **platform migration**. In Kyndryl's 2025 survey, 98 percent of organizations were moving at least some applications off the mainframe,¹⁵ and every batch job that moves needs a sort on the other side. The third is the **support model**. Three of the four groups above have changed owner or name since 2018, and a new owner can change release schedules and support terms. The fourth is **consolidation**. One sort on z/OS, another in a rehosting runtime and a third on distributed servers means three sets of skills and contracts to keep up, and three products that each behave a little differently.

None of these reasons says anything about product quality. The products in place today are mature and well documented, and the people who support them know mainframe batch inside out. The burden of proof is on the change.

2. Where sort gets called from

A replacement has to answer on every route your batch uses to reach the sort product. IBM documents three for DFSORT: a JCL EXEC statement naming the program (PGM=ICEMAN or PGM=Sort, for example), a COBOL or PL/I program using the language's own sort, and an assembler program using a system macro instruction.¹⁶ Precisely describes the same three for Syncsort MFX.¹⁷ Language Environment adds a callable service, CEE3SRT, which calls DFSORT and supports only the E15, E35 and E32 exits.¹⁸

Exhibit 1. How batch reaches the sort, and how a replacement goes in behind each route

Route	How the sort is reached	How a replacement goes in	Watch for
JCL step	EXEC PGM=SORT, ICEMAN, SYNC SORT or a product name; SORTIN, SORTOUT, SYSIN and SORTWKnn DD statements	Library search order on z/OS; program mapping in a rehosting runtime; symbolic link or wrapper under the old name on Linux, UNIX or Windows	STEPLIB or JOBLIB pointing at the old product's libraries; PARM strings
Tool front end	EXEC PGM=ICETOOL or SYNCTOOL; IEBGENER redirected to ICEGENER	Same name resolution as above	Operator coverage; return codes used for flow control; copy steps that don't look like sorts
COBOL and PL/I	SORT and MERGE statements; PLISRTx calls; input and output procedures run as E15 and E35 exits	Name resolution on z/OS; a compiler directive on some distributed COBOL compilers	Exit linkage and return codes; SORT-RETURN checks; recompiling
Assembler and callable services	LINK or ATTACH with a parameter list; Language Environment CEE3SRT	Name resolution; a compatible parameter list	Parameter-list formats; exits other than E15 and E35
Other vendors' software	Database utilities, report writers and similar products that sort internally	Often can't be replaced: the product may call a specific sort by a private name	Db2 utilities call DFSORT through DFSORT-only aliases
Distributed scripts and APIs	Shell or batch scripts calling a sort command; C or Java calls	Symbolic link, wrapper or search-path order; library substitution for APIs	Different command syntax; exit-status conventions

Sources: IBM DFSORT Application Programming Guide; IBM II14047; Precisely Syncsort MFX Programmers Guide; IBM Language Environment; Micro Focus documentation.^{16, 5, 17, 18, 19}

The name SORT doesn't belong to any one product. On z/OS the module that answers is whichever one the system finds first in its search order: job or step libraries, then the link pack area and link list. IBM's advice for shops whose main sort isn't DFSORT is to put DFSORT later in that order,⁵ and Precisely's knowledge base has an article on the opposite problem, DFSORT getting invoked instead of Syncsort MFX.²⁰ The same idea applies elsewhere. A rehosting runtime maps a JCL program name to an executable, and its documentation tells you which names its bundled sort answers to.^{11, 12} On Linux or UNIX it's a symbolic link or a wrapper under the old command name. For COBOL compiled off the mainframe, the route is set at compile time. Micro Focus's CALLSORT directive, in its documentation's words, "Defines the program to be called to handle all SORT and MERGE operations."¹⁹

Two routes get missed a lot. IBM recommends ICEGENER in place of IEBGENER,²¹ and where IEBGENER has been redirected to it, ordinary copy steps depend on the sort product without ever naming it. Other vendors' software may also sort internally, calling a specific product by a private name.

3. How the dialects differ

The mainframe control statements (SORT, MERGE, INCLUDE, OMIT, INREC, OUTREC, OUTFIL, SUM and the rest) are shared across DFSORT, Syncsort MFX and CA Sort closely enough that most job streams are portable, and the products have long competed partly on accepting each other's statements. Each vendor has also extended the language on its own schedule. An IBM technote for customers converting from Syncsort says DFSORT has no equivalent for a Syncsort syntax of the form L(C'...').²² The tool front ends differ in name and scope. DFSORT's ICETOOL has 17 operators;²³ Syncsort's counterpart is SYNCTOOL.²⁴

The emulations cover documented subsets. AWS lists six ICETOOL operators as supported in its runtime,¹² and Micro Focus's 2008 documentation for MFJSORT said exits other than E15 and E35 were not supported.¹¹ That isn't a criticism. Each vendor documents its scope. So the question to ask is which statements, operands, options and exits

your shop actually uses.

The differences that are hardest to spot sit outside the control statements. DFSORT's EQUALS option, which decides whether records with identical keys keep their input order, is “usually the installation default”,¹⁶ so the same statements can behave differently on two systems running the same product. Products can also differ by design. Micro Focus staff explained in 2013 that MFSORT returns duplicates in input order “regardless of EQUALS=NO/YES”.²⁵ Defaults for empty inputs, summary overflow and abending work the same way (Section 4). They're part of the dialect, and they live in the installation options, where no job will show them to you.

Off the mainframe the question changes. Syncsort's distributed products, now Connect ETL, use a completely different language, built from slash-prefixed options like /INFILE.⁸ Replacing Syncsort on UNIX or Windows is its own problem, because a replacement has to read that syntax or convert it. Changing platforms also brings in data representation (EBCDIC and ASCII collation, packed and zoned decimal, record descriptor words, VSAM, Unicode and double-byte data), and each of those needs its own test. Exhibit 2 shows where the risk is.

Exhibit 2. Where the compatibility risk is, by feature family

Feature family	Examples	Risk	What to check
Core ordering and selection	SORT, MERGE, COPY, INCLUDE, OMIT, SUM, RECORD	Low	Order of equal keys; SUM overflow handling; alternate collating sequences
Basic reformatting	INREC, OUTREC, OUTFIL with field moves, constants and edit masks	Low to medium	Less common operands; padding and truncation of output length
Extended functions	Conditional reformatting, find and replace, parsing, date arithmetic, reports	Medium to high	Vendor-specific keywords and constants; check every operand you use, one by one
Joins	JOINKEYS, JOIN, REFORMAT	Medium	Unpaired-record handling; order of joined output
Tool front ends	ICETOOL and SYNCTOOL operators	High	Operator coverage; report layouts; return codes
Symbolic names	SYMNAMES and field dictionaries	Medium	Symbol file format and location
User exits	E15, E35 and other exit points in COBOL, assembler or C	High	Supported exit points, languages and return codes; recompiling for a new platform
Installation defaults	EQUALS, NULLOUT, OVFO, ABEND or NOABEND, message level	High	Pull the options in force and set the new product to match
Operational interface	Return codes, message IDs, SMF records, listings	High	Scheduler conditions, automation rules, reports built on SMF data
Data representation	EBCDIC and ASCII, packed and zoned decimal, VB records, VSAM, Unicode, DBCS	High on a new platform	Collation, sign handling, record formats, unsupported file types
Distributed Syncsort syntax	Slash-option task definitions (/INFILE and similar)	Separate language	Direct support or conversion; scripts that build commands on the fly

The risk ratings are my own judgment, for planning. They aren't measurements.

4. Return codes, messages and output order

Schedulers don't read sort control statements. They read return codes. Automation reads message IDs, and downstream jobs read the output, including the order it's in. That interface binds you just as much as the program name does.

DFSORT returns 0 on success. It can be told to return 4 for conditions like an empty output or summary-field overflow, and it returns 16 on failure when it's set not toabend.¹⁶ IBM also documents 20 and 28 for some unsuccessful completions.²⁶ ICETOOL has its own scale, from 0 to 24, and its COUNT operator exists mostly to set a return code for a later step to test.²³ Messages carry product-specific prefixes: ICE for DFSORT and ICETOOL, WER for Syncsort MFX, whose messages look like WERnnnx with a suffix letter giving the class.²⁷ Even exits have conventions. Precisely documents 20 as a valid exit return code only for COBOL or C E15 and E35 exits.²⁷

Here's an example: one empty file, three outcomes

Let's say a copy step's input happens to be empty one night. Under DFSORT, what happens depends on the NULROUT option. With RCO the step issues message ICE173I and ends with return code 0. With RC4 it issues the same message and ends with 4. With RC16 it issues ICE206A, terminates and ends with 16.¹⁶ A following step coded to run only when the return code is below 4 runs in the first case and gets skipped in the other two.

ICETOOL users often make the test explicit. COUNT FROM(IN5) EMPTY issues ICE646A and sets return code 12 when the data set is empty.²³

So: a replacement that handles the empty file correctly and returns a different code has changed your batch schedule. Your installation's setting is the spec, whatever the product's default happens to be.

Messages matter the same way. Automation rules, runbooks and job-log scanners key on message IDs, and some parse the message text for record counts. New IDs mean finding and changing those rules. Capacity and chargeback reports built on a product's own SMF records (DFSORT's is type 16)²⁸ will need a new source.

5. Proving it matches, and how fast it runs

The standard of proof is the same outputs, the same return codes and acceptable run times on your own work. The method below applies to any replacement, including a move from one mainframe product to another.

- 1 Take a census.** Scan JCL, procedures, control-statement libraries, program source and scripts for every invocation. Pull out every statement, operand, option and exit used. Join that to the SMF step records (types 16 and 30 on z/OS) to see what actually runs, and how often. The census is your spec. A vendor's feature list isn't.
- 2 Build the test set from production.** Pick statements that cover every feature in the census, with real data or data that looks like it: key distributions, packed fields, variable lengths, empty files.
- 3 Compare outputs byte for byte, and explain every difference.** Where NOEQUALS is in effect, equal-keyed records can legitimately come out in a different order. Either run the comparison with EQUALS or use a tool that treats each group of equal keys as a set. Mask run dates and times in report headings.
- 4 Test the failure paths.** Empty inputs, summary overflow, invalid packed data, running out of work space and missing DD statements should all give the return codes the scheduler expects.
- 5 Measure at volume, in parallel.** Sort behaves differently once the data outgrows memory, and again when a lot of sorts are competing for memory. Measure elapsed time, CPU and I/O, and judge the result on the critical path of the batch window. Step-by-step numbers can fool you.
- 6 Tune both sides.** Compare a tuned incumbent with a tuned candidate on the same hardware and data. Treat benchmark claims from any vendor as a place to start asking questions. They aren't evidence.

6. Read the contracts first

The technical work takes months. Notice periods and renewal dates are fixed, and they often fall at awkward times. There are two contracts in play, the one you're leaving and the one you're signing, and somebody should read both before the project plan gets drawn up. Exhibit 3 lists the questions. The answers belong with procurement and your lawyers.

Exhibit 3. Contract checklist for both sides of the change

Topic	Outgoing agreement	Incoming agreement
Term and notice	Renewal date, notice period and form, auto-renewal	Initial term; renewal and charge-change terms
Transition use	Right to run the current version through parallel running	Non-production use before production starts
License scope	Machines, partitions and recovery systems covered	Metric for test, recovery, containers, cloud, outsourcers
Audit	Audit terms; certification of removal	Proportionate audit clause; self-certification
Support	Fixes through the notice period	Severities, response times, hours, supported releases
Continuity	Rights that must survive termination	Source escrow and release triggers; change of control
Exit	Return or destruction of software and keys	Exit assistance; production data in support cases
Risk allocation	Liabilities that survive	Liability cap, IP indemnity, warranty of documented function

A general checklist for planning. It isn't legal advice.

Watch your audit exposure, because a transition is when it's easiest to go over your entitlement: two products installed, and more test systems than usual. In Flexera's 2026 survey of more than 500 IT asset management professionals (a vendor-sponsored study), 48 percent of organizations said they had been audited in the previous year.²⁹ For EU financial entities, the Digital Operational Resilience Act, which has applied since January 17, 2025, sets minimum content for contracts covering ICT services. That includes service-level descriptions, termination rights with minimum notice periods and, for critical or important functions, exit strategies with a mandatory adequate transition period.³⁰ Whether a sort license is in scope is a question for your lawyers. The checklist holds up either way.

7. A phased plan

Exhibit 4 puts the technical and commercial work in one sequence. Two things matter most. You make the switch by changing name resolution one application group at a time, so falling back is the same change in reverse and doesn't touch any JCL or program. And you keep the old product installed and licensed until the new one has run cleanly through month-end, quarter-end and, ideally, year-end.

Exhibit 4. A phased replacement plan

Phase	Main work	Done when
0. Prepare	Read both contracts; put the notice dates on the calendar; get parallel-running rights	Commercial timetable agreed
1. Census	Invocations, statements, options, exits, return codes, defaults	Census signed off
2. Gap analysis	Each gap: vendor change, rewrite, or keep on the old engine	Every gap closed or accepted
3. Prove	Test set, byte comparison, failure paths, volume tests	Test report signed off
4. Parallel run	Both engines on production inputs; compare every cycle	Clean month- and quarter-end
5. Cut over in waves	Switch name resolution per application group	Fallback unused for agreed period
6. Decommission	Remove old libraries; give notice; certify removal; archive evidence	License position documented

A rough sequence. How long it takes depends on the size of the census and on the contract dates.

8. Bottom line

- **Syntax is only the start.** Most of the differences are in extensions and exits. The installation defaults are the other big one.
- **The census is the spec.** What your shop actually uses decides whether this is feasible. Vendor feature lists and compatibility percentages can't tell you that.
- **Keep the name, prove the interface.** Show that return codes, messages and output order match.
- **Buy time before you buy software.** Notice dates and parallel-running rights set the real timetable.
- **Staying put is a legitimate outcome.** Even then, the census pays for itself at renewal.

About the author

Bill Ahlbrandt is President and CEO of Ahlbrandt Software, which he founded in 1993. Before that he was a systems programmer, Director of Information Security and Director of R&D at Acxiom. He has been working on mainframe sort and batch for more than thirty years, and his company's sort engine runs mainframe sort jobs on Linux and other platforms for banks, insurers and government agencies.

Full disclosure: my company sells sort software for platforms other than the mainframe, and it reads mainframe sort control statements. Everything in this paper applies to any replacement, including a switch between the mainframe products themselves.

References

1. Wikipedia, "Mainframe sort merge", accessed September 2026. Secondary source.
2. Wikipedia, "Computer Associates", accessed September 2026. Secondary source.
3. IBM Support, "DFSORT" product page, updated 1 October 2025.
4. IBM, *z/OS Version 2 Release 1 Licensed Program Specifications*, program number 5650-ZOS (list of optional priced features).
5. IBM, Informational APAR I114047, "Use of DFSORT by DB2 Utilities", last modified 23 January 2023.
6. Precisely, press release announcing the Syncsort rebrand to Precisely; reported by destinationCRM, 14 May 2020. Vendor figures.
7. Precisely, "Syncsort MFX" product page, accessed September 2026. Vendor source.
8. Precisely, *Connect ETL 9.13 DTL Guide*, "/INFILE, /FILE"; Precisely Support, Connect ETL (formerly DMExpress).
9. Broadcom TechDocs, "CA Sort", in the Legacy Bookshelves and PDFs collection, accessed September 2026.
10. Rocket Software, press release on closing the acquisition of OpenText's AMC business, Business Wire, 30 April 2024.
11. Micro Focus, Mainframe Subsystem documentation, "Utility Programs" (MFJSORT), © 2008, and "Mfsort Utility", © 2009.
12. Amazon Web Services, *AWS Mainframe Modernization User Guide*, "Sort Utilities", accessed September 2026.
13. IRI, "CoSort Translates JCL Sort Steps into SortCL Scripts", product page, accessed September 2026. Vendor source.
14. GnuCOBOL project, GCSORT, feature request #400, updated 23 February 2023.
15. Kyndryl, State of Mainframe Modernization Survey, 9 September 2025 (n = 500). Vendor-sponsored.
16. IBM, *z/OS DFSORT Application Programming Guide*, SC23-6878 (V2R1, V2R2): invocation; E15 exits; PARM options.
17. Precisely, *Syncsort MFX 3.1 Programmers Guide*, "An Introduction to Syncsort MFX".
18. IBM, *z/OS Language Environment Programming Reference*, "CEE3SRT: call DFSORT".
19. Micro Focus, Visual COBOL 6.0 documentation, "CALLSORT" compiler directive.
20. Precisely Customer Community, "DFSORT Invoked Instead of Syncsort MFX for z/OS During System Sort Execution".
21. IBM, *z/OS 3.1 documentation*, performance topic "Use ICEGENER instead of IEBGENER".
22. IBM Support technote, "Converting from SYNCSORT to DFSORT / DFSORT does not have an equivalent of L(C'???)", 23 June 2026.
23. F. L. Yaeger, *DFSORT: ICETOOL Mini-User Guide*, IBM DFSORT Team, October 2010.
24. Precisely Customer Community, knowledge article "Implementing SYNCTOOL in Syncsort MFX" (title as listed).
25. Rocket Community, "What is the option in MFSORT for STABLE/NOSTABLE or EQUALS/NOEQUALS", 15 February 2013.
26. IBM, APAR PI40493 (DFSORT invoked with NOABEND returns 16, 20 or 28), closed 16 July 2015.
27. Precisely, *Syncsort MFX 3.1 Programmers Guide*, "Syncsort MFX Messages".
28. IBM, DFSORT SMF type 16 record documentation and reporting samples, 15 August 2022.
29. Flexera, *2026 State of ITAM Report*, 24 June 2026 (n > 500). Vendor-sponsored.
30. Regulation (EU) 2022/2554 (Digital Operational Resilience Act), Articles 28, 30 and 64; applicable from 17 January 2025.