

The Mainframe Exit Playbook

The worksheets, checklists and gate criteria we use before anyone touches a line of COBOL

Written for the people who have to answer for the mainframe: what to count before you ask for a quote, which path each workload should take, how a modernization programme is sequenced, and what has to be true before a workload is allowed to cut over. Fill it in. It is worth more completed than read.

Why the mainframe question changed

The mainframe was never the problem. The problem was always the cost of finding out what it does. A traditional modernization programme spent its first year on discovery: reading undocumented COBOL, reconstructing what each job was for, and hand-writing a regression suite that had never existed. That phase carried most of the budget and all of the risk, which is why so many business cases were declined twice and shelved.

Agentic tooling changed the economics of that phase specifically. AWS Transform, AWS's agentic service for mainframe work, analyses an uploaded codebase, documents what each program does, extracts business rules with a trace back to the exact source file and line, plans migration waves, translates COBOL to Java, and generates tests. AWS publishes the mainframe modernization agent as free; what a partner is paid for is the engineering around it.

What did not change is who is accountable when a batch job produces a different number. Generation is cheap now. Verification is the work.

What this playbook is for

- Counting your estate accurately enough that any quote you receive is comparable.
- Assigning each workload to one of six honest paths, including the two that mean "do not move it".
- Sequencing waves so the first one proves the pattern instead of betting the programme.
- Defining the gate criteria a workload must pass before cutover, performance included.

One rule underneath all of it. Nothing scales before it is measured. A pilot workload running in parallel with the mainframe on production data tells you more than any vendor projection, including ours.

Estate inventory

Fill this in before requesting a proposal from anyone. Every serious estimate depends on these numbers, and a vendor who quotes without them is quoting a template.

WHAT TO COUNT	YOUR NUMBER	WHERE TO FIND IT
Total lines of COBOL (excluding comments)		Source library scan
Number of distinct programs		Source library
Copybooks		Source library
JCL jobs, and how many run nightly		Scheduler
CICS transactions		CSD
DB2 tables / VSAM files		Catalog
Other languages present (PL/I, RPG, Assembler, Natural)		Source library
Batch window: start, end, current finish time		Scheduler history
Peak online transaction rate		CICS statistics
Annual MIPS / hardware / licence cost		Finance
People who can still read this code, and their retirement horizon		You already know
Regulatory constraints on where this data may run		Compliance

The two numbers people forget. The batch window is the real constraint on almost every mainframe programme, and the retirement horizon is the real deadline. Neither appears on a technical inventory unless you put it there.

Six paths, and how to choose

No estate takes one path. Assign a path per workload, write down the reason, and keep the reason where the next person can read it.

PATH	CHOOSE IT WHEN	WHAT IT COSTS YOU
Refactor	The business logic is still correct and the estate is large, stable, batch-heavy. COBOL is translated to Java, behaviour held identical.	Least design risk, largest test burden.
Reimagine	The logic is worth keeping but the structure is not. Rules are extracted, then the workload is rebuilt cloud-native from those requirements.	More design work, better end state.
Replatform	The driver is exiting hardware and licences, not improving code. The workload moves to a cloud runtime with minimal change.	Fast, but the debt moves with it.
Replace	A mature product already does this. Payroll and core administration workloads often have a SaaS equivalent that beats any rewrite.	Migration of data and process, not code.
Retire	Nobody has run it in years. Most estates carry more of this than anyone expects.	Only the effort to prove it is truly unused.
Retain	It should stay for now, integrated rather than moved. A plan that says this about nothing is a sales plan.	Integration work, and honesty.

Assign paths here

WORKLOAD	PATH	BUSINESS CRITICALITY	WHY THIS PATH

AWS Transform readiness

What has to be in place before agentic analysis is worth running. Most of it is access and inventory, and most delays are here rather than in the engineering.

Source and data

- ☐ Complete source extract: programs, copybooks, JCL, procs and parameter cards
- ☐ CICS CSD and IMS definitions where they apply
- ☐ DB2 DDL and VSAM record layouts
- ☐ Confirmation that the extract is the version actually running in production
- ☐ Test data, or an agreed way to generate representative data

Environment and access

- ☐ An AWS account for the work, with IAM boundaries agreed
- ☐ Data residency and retention policy confirmed for that account
- ☐ Security review of the service's data handling completed with your team
- ☐ Named owner on your side for questions the code cannot answer

Facts nobody can look up for you

- ☐ The batch window, as a hard time, and what happens if it is missed
- ☐ Peak transaction volumes and the response times users actually expect
- ☐ Which workloads a regulator must be consulted about before they move
- ☐ Which upstream and downstream systems break if a schema changes
- ☐ Who operates the mainframe today, and what their role is afterwards

Coverage limit worth knowing early. AWS Transform targets COBOL for transformation. PL/I, RPG, Assembler and Natural/Adabas need a different approach and should be priced separately rather than folded into the same line.

Wave planning

Sequence on business priority and blast radius, not on what looks easiest. The first wave exists to prove the pattern: pick something real enough to be convincing and contained enough to be reversible.

WAVE	WORKLOADS	PATH	DEPENDENCIES TO CLEAR FIRST	ROLLBACK PLAN
Pilot				
1				
2				
3				
4				

Sequencing rules that hold up

- A wave that cannot be rolled back is not a wave, it is a cutover.
- Move data and the code that owns it in the same wave, or accept a synchronization problem instead.
- Anything shared by many workloads either goes first or goes last. Nothing in between survives contact.
- Leave the workload with the strictest regulator until the pattern has been proven twice.
- Every wave ends with a named engineer signing it off. If nobody signs, nothing shipped.

Cost model

Two halves: what you stop paying, and what the work costs. The first half is easier than people expect and is usually the half that carries the business case.

What you stop paying, per year

LINE	ANNUAL COST TODAY
Hardware, capacity and maintenance	
Software licences tied to the platform	
Specialist staffing and contractor cover	
Disaster recovery capacity	
Deferred change: work not done because the system could not take it	
Total	

What the work costs

- **Assessment.** Fixed price, fixed scope. Ours is \$15,000 over three weeks for roughly a million lines or ten applications, and the output is yours to use with any partner.
- **Pilot.** One workload end to end, including parity testing on production data. Ours starts at \$60,000.
- **Waves.** Priced per wave against the plan. The honest drivers are estate size, data complexity, integration count and how much test data already exists.
- **Tooling.** The AWS Transform mainframe agent is published by AWS as free. Budget for the cloud run-cost of the target environment instead.
- **Parallel running.** Both systems run at once for part of the programme. Plan for it rather than discovering it.

Model the run-cost side first. Our cloud migration calculator produces a five-year run-cost projection in a few minutes: migration.dedicatt.com. It does not price the transformation — that comes out of the assessment — but it sizes the half of the case that is about what you stop paying.

Cutover gate criteria

Agree these before the wave starts, in writing. A gate agreed afterwards is a negotiation, and the pressure at that point is always in one direction.

Functional parity

- ☐ Characterization tests captured the current behaviour before any change
- ☐ Modernized workload matches the mainframe on production data, not sample data
- ☐ Every extracted business rule traces to a source file and line
- ☐ Differences are enumerated and each one has a written, accepted explanation

Performance parity

- ☐ Batch completes inside the real window, measured on real volume
- ☐ Peak transaction response times at or under the current baseline
- ☐ Behaviour under the worst historical load, not just an average day
- ☐ Month-end and year-end cycles rehearsed if they fall inside the horizon

Operational readiness

- ☐ Runbooks written with the operators who will use them
- ☐ Monitoring and alerting in place before traffic moves, not after
- ☐ Rollback path tested, not merely documented
- ☐ Your operations team trained on the new stack for this wave
- ☐ A named engineer has signed the wave off

Evidence

- ☐ Parity runs recorded with inputs, outputs and timestamps
- ☐ Audit trail of what each agent produced and from which source lines
- ☐ Sign-off record retained where an auditor can find it

What to do with this

If you fill in the inventory and the path assignments, you can put the same question to three vendors and compare the answers, which is not true today for most mainframe programmes. That is the point of publishing it.

If you want a second opinion on what you filled in

Our readiness assessment takes three weeks at a fixed \$15,000 and produces the full inventory, a dependency map across the estate, a path assigned to every workload with the reasoning, a sequenced wave plan, and a fixed-price proposal for what follows. All of it is yours to keep and to use with any partner, including none of us.

What we are, plainly

Dedicatted is an AWS Premier Tier Services Partner and the only AWS Generative AI and MSP partner in Canada. We build our mainframe practice on AWS Transform rather than proprietary tooling, because a service AWS publishes and documents is one your team can verify independently. We have no published mainframe case study yet, which is exactly why the assessment is structured so you keep the output whatever you decide next.

Talk to an engineer. A senior engineer, not a sales rep, replies within one business day: dedicatted.com/services/engineering/mainframe-modernization · contact@dedicatted.com

Dedicatted

AWS Premier Tier Services Partner
Toronto · Cracow · Lviv

August 2026

dedicatted.com