CodeProva 01 Nov 2026

CardDemo

Modernization plan for Northwind Card Services

We recommend converting 27 programs directly with AI-assisted translation and 14 more with a business expert confirming their rules first. Two programs should stay in place behind an interface for now and one program appears to be unused.

Prepared by **CodeProva** Reference A-6d22ed24e4 01 Nov 2026 Confidential

The recommended path Details



Convert	27
Convert with SME review	14
Wrap and defer	2
Confirm and retire	1

Start here 545 lines

CBTRN03C

Print the transaction detail report

Run the legacy and new versions against the same production-like input and compare TRANREPT record by record; every difference is explained or fixed.

44 programs and 22,893 lines reviewed

16 business rules found in the code

16 risks, each with an owner

4 items we need from your team

Memo A-6d22ed24e4

To	Northwind Card Services
From	CodeProva
Subject	How we recommend modernizing CardDemo
Reference	A-6d22ed24e4

We reviewed 44 COBOL programs (22,893 lines of code) along with 65 copybooks, 56 batch jobs and 25 online transactions. Nothing was executed: every observation below comes from reading the source, and each one links to the file and line it is based on.

We recommend converting 27 programs directly with AI-assisted translation and 14 more with a business expert confirming their rules first. Two programs should stay in place behind an interface for now and one program appears to be unused.

We suggest starting with CBTRN03C (print the transaction detail report). At 545 lines it can be finished quickly, and it is a batch program, so the new version can be proven by running both on the same input and comparing the output record by record.

We identified 16 business rules in the code — for example, in CBACT04C: when interest rate is not 0, compute interest; compute fees. These rules become the reference for testing and should be confirmed by the people who own them.

The main risks are that some logic paths are missed or reordered in conversion, that conversion is blocked until the data model is decided and that untested paths behave differently after conversion. Each is in the risk register with a suggested owner and how we would manage it.

— CodeProva

Rules your code enforces All 16

When interest rate is not 0
compute interest; compute fees · CBACT04C

When approved auth count is at most 0 and approved auth count is at most 0
delete auth summary · CBPAUP0C

When diff is at least expiry days
mark as “qualified for delete” · CBPAUP0C

When account credit limit is at least temp balance
otherwise set validation fail reason to 102 · CBTRN02C

Roadmap each stage starts when the previous one is signed off

0

Before scoping with your team

Close the gaps

Settle the open questions so these programs can be planned or removed from scope.

- CBTRN01C

What is needed
See “What we need from you”.

1

1 program
545 lines

Pilot

Prove the method end to end on one contained program, with full evidence.

- CBTRN03C

Ready to start when
- A masked copy of recent production input files for comparison runs.
- One business expert for a one-hour rule walkthrough.

Proven by
Run the legacy and new versions against the same production-like input and compare TRANREPT record by record; every difference is explained or fixed.

2

11 programs
2,955 lines

Batch foundations

Move the remaining low-risk batch processing, where equivalence is easiest to prove.

- CBACT01C
- CBACT02C
- CBACT03C
- CBACT04C
- CBCUS01C
- CBEXPORT
- CBIMPORT
- CBTRN02C
- COBSWAIT
- COBTUPDT
- CSUTLDTC

Ready to start when
- Pilot signed off.
- Target file and table formats agreed.

Proven by
Parallel runs: legacy and new jobs process the same daily files; outputs are compared automatically.

3

15 programs
5,695 lines

Online screens and services

Replace terminal screens with services and a web front end, one business function at a time.

- COACTVWC
- COADM01C
- COBIL00C
- COCRDSLC
- COMEN01C
- COPAUS2C
- CORPT00C
- COSGN00C
- COTRN00C
- COTRN01C
- COTRN02C
- COUSR00C
- COUSR01C
- COUSR02C
- COUSR03C

Ready to start when

- Front-end approach agreed (web UI, API-only, or both).
- Shared data used by batch already migrated or synchronised.

Proven by

Recorded screen journeys replayed against both versions; responses and data changes compared.

4

14 programs
9,144 lines

Expert-reviewed core

Convert the programs that carry sensitive rules or unusual constructs, with SME confirmation.

- CBPAUP0C
- CBSTM03A
- CBSTM03B
- COACCT01
- COCRDLIC
- COCRDUPC
- CODATE01
- COPAUS0C
- COPAUS1C
- COTRTLIC
- COTRTUPC
- DBUNLDGS
- PAUDBLOD
- PAUDBUNL

Ready to start when

- Business rules catalog reviewed and signed off by owners.
- Target data model for the IMS database decided.

Proven by

Rules confirmed by a named expert before translation; characterisation tests for every rule in the catalog; output comparison as above.

5

2 programs
4,139 lines

Wrap and decide later

Keep these running behind a service interface so the rest of the estate can move around them.

- COACTUPC
- COPAUA0C

Ready to start when

- Interface design for each wrapped program.

Proven by

Interface contract tests ensure callers see identical behaviour through the wrapper.

What a change affects

Choose a program or data store to see everything that would need retesting.

Example: the program with the widest reach in this estate

`CBTRN02C`
Recommendation: Convert

13 things to retest if this changes

Programs that call it (0)

None

Programs that use data it writes (12)

- CBACT01C
- CBACT04C
- CBEXPORT
- COACCT01
- COACTUPC
- COACTVWC
- COBIL00C
- COPAUA0C
- COPAUS0C
- COTRN00C
- COTRN01C
- COTRN02C

Batch jobs that run it

POSTTRAN

Online transactions that reach it

None

It calls

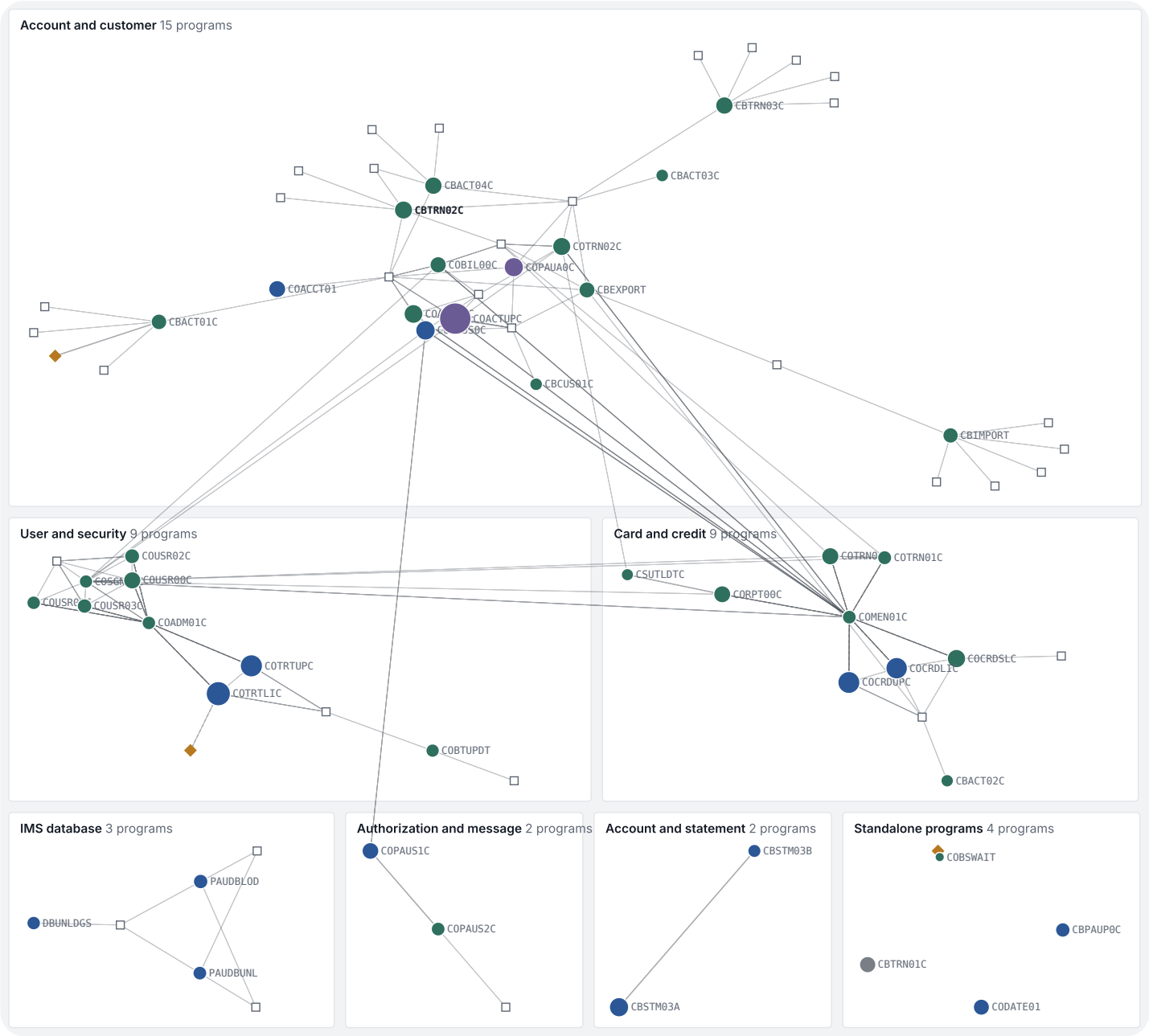
None

It reads

- AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS
- AWS.M2.CARDDEMO.DALYTRAN.PS

It writes

- AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS
- AWS.M2.CARDDEMO.DALYREJS
- AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS
- AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS



Top risks Register

High	**Unstructured jumps between sections of logic**	Engineering
High	**Depends on a hierarchical IMS database**	Data architecture
High	**Many decision paths in one program**	QA
High	**Very large single program**	Engineering
High	**Program changes its own control flow at run time**	Engineering lead with business SME

What we need from you

- [ ] **Send the source for COCRDSEC**
  It is called by programs in scope but was not in the code we received.
- [ ] **Check run history for CBTRN01C**
  We found nothing that runs it. If it has not run in the last year, it can leave scope.
- [ ] **Nominate a data architect for the IMS database**
  Its target data model decides how the IMS programs are converted.
- [ ] **Name a business owner for each capability**
  They confirm the business rules catalog, which is the reference for testing.

The recommended path

One recommendation per program; the reasons are in the program profiles.

Convert

27 programs · 9,195 lines

Translate with AI; prove equivalence with old-versus-new testing.

- CBACT01C
- CBACT02C
- CBACT03C
- CBACT04C
- CBCUS01C
- CBEXPORT
- CBIMPORT
- CBTRN02C
- CBTRN03C
- COACTVWC
- COADM01C
- COBIL00C
- COBSWAIT
- COBTUPDT
- COCRDSLC
- COMEN01C
- COPAUS2C
- CORPT00C
- COSGN00C
- COTRN00C
- COTRN01C
- COTRN02C
- COUSR00C
- COUSR01C
- COUSR02C
- COUSR03C
- CSUTLDTC

Convert with SME review

14 programs · 9,144 lines

Translate, but a business expert confirms the rules before sign-off.

- CBPAUP0C
- CBSTM03A
- CBSTM03B
- COACCT01
- COCRDLIC
- COCRDUPC
- CODATE01
- COPAUS0C
- COPAUS1C
- COTRTLIC
- COTRTUPC
- DBUNLDGS
- PAUDBLOD
- PAUDBUNL

Wrap and defer

2 programs · 4,139 lines

Leave in place behind a service interface; revisit once its dependants have moved.

- COACTUPC
- COPAUA0C

Confirm and retire

1 program · 415 lines

No caller found; confirm with run history, then drop it from scope.

- CBTRN01C

Business capabilities

Programs grouped by the work they do, a natural unit to modernize and test together.

Convert · Convert with SME review · Wrap and defer · Confirm and retire

Account and customer

- **CBACT01C** — READ THE ACCOUNT FILE AND WRITE INTO FILES
- **CBACT03C** — Read and print account cross reference data file
- **CBACT04C** — This is a interest calculator program
- **CBCUS01C** — Read and print customer data file
- **CBEXPORT** — Export Customer Data for Branch Migration
- **CBIMPORT** — Import Customer Data from Branch Migration Export
- **CBTRN02C** — Post the records from daily transaction file
- **CBTRN03C** — Print the transaction detail report
- **COACCT01** — Online program
- **COACTUPC** — Accept and process ACCOUNT UPDATE
- **COACTVWC** — Accept and process Account View request
- **COBIL00C** — Bill Payment - Pay account balance in full and a
- **COPAUA0C** — Card Authorization Decision Program
- **COPAUS0C** — Summary View of Authoriation Messages
- **COTRN02C** — Add a new Transaction to TRANSACT file

Card and credit

- **CBACT02C** — Read and print card data file
- **COCRDLIC** — List Credit Cards
- **COCRDSLC** — Accept and process credit card detail request
- **COCRDUPC** — Accept and process credit card detail request
- **COMEN01C** — Main Menu for the Regular users
- **CORPT00C** — Print Transaction reports by submitting batch
- **COTRN00C** — List Transactions from TRANSACT file
- **COTRN01C** — View a Transaction from TRANSACT file
- **CSUTLDTC** — Subprogram program

User and security

- **COADM01C** — Admin Menu for Admin users
- **COBTUPDT** — Update Transaction type based on user input
- **COSGN00C** — Signon Screen for the CardDemo Application
- **COTRTLIC** — List Transaction Type for updates and deletes
- **COTRTUPC** — Accept and process TRANSACTION TYPE UPDATE
- **COUSR00C** — List all users from USRSEC file
- **COUSR01C** — Add a new Regular/Admin user to USRSEC file
- **COUSR02C** — Update a user in USRSEC file
- **COUSR03C** — Delete a user from USRSEC file

IMS database

- **DBUNLDGS** — Batch program
- **PAUDBLOD** — Batch program
- **PAUDBUNL** — Batch program

Account and statement

- **CBSTM03A** — Print Account Statements from Transaction data
- **CBSTM03B** — Does file processing related to Transact Report

Authorization and message

- `COPAUS1C` Detail View of Authorization Message
- `COPAUS2C` Mark Authorization Message Fraud

Standalone programs

- `CBPAUP0C` Delete Expired Pending Authoriation Messages
- `CBTRN01C` Post the records from daily transaction file
- `COBSWAIT` UTILITY PROGRAM TO WAIT (PARM IN CENTISECONDS)
- `CODATE01` Online program

Business rules found in the code

Once their owners confirm them, they become the tests the new system must pass.

`CBACT04C` — This is a interest calculator program

When	Then	Type	Source	Owner confirms
Interest rate is not 0 `DIS-INT-RATE NOT = 0`	compute interest; compute fees	Money	`CBACT04C.cbl:214`	☐

`CBPAUP0C` — Delete Expired Pending Authoriation Messages

When	Then	Type	Source	Owner confirms
Approved auth count is at most 0 and approved auth count is at most 0 `PA-APPROVED-AUTH-CNT <= 0 AND PA-APPROVED-AUTH-CNT <= 0`	delete auth summary	Validation	`CBPAUP0C.cbl:156`	☐
Diff is at least expiry days `WS-DAY-DIFF >= WS-EXPIRY-DAYS`	mark as “qualified for delete”	Timing	`CBPAUP0C.cbl:284`	☐

`CBSTM03A` — Print Account Statements from Transaction data

When	Then	Type	Source	Owner confirms
Xref card number equals card num `XREF-CARD-NUM = WS-CARD-NUM (CR-JMP)`	—	Validation	`CBSTM03A.CBL:420`	☐

`CBTRN02C` — Post the records from daily transaction file

When	Then	Type	Source	Owner confirms
Account credit limit is at least temp balance `ACCT-CREDIT-LIMIT >= WS-TEMP-BAL`	otherwise set validation fail reason to 102	Money	`CBTRN02C.cbl:407`	☐
Account expiration date is at least dalytran original ts `ACCT-EXPIRAION-DATE >= DALYTRAN-ORIG-TS (1:10)`	otherwise set validation fail reason to 103	Timing	`CBTRN02C.cbl:414`	☐
Dalytran amount is at least 0 `DALYTRAN-AMT >= 0`	add dalytran amount to account current cyc credit	Money	`CBTRN02C.cbl:548`	☐

`CBTRN03C` — Print the transaction detail report

When	Then	Type	Source	Owner confirms
Current card number is not transaction card number `WS-CURR-CARD-NUM NOT= TRAN-CARD-NUM`	—	Validation	`CBTRN03C.cbl:181`	☐

`COACTUPC` — Accept and process ACCOUNT UPDATE

When	Then	Type	Source	Owner confirms
The record is marked “valid general purp code” `VALID-GENERAL-PURP-CODE`	otherwise mark as “input error”; mark as “edit us phonea not ok”	Status	`COACTUPC.cbl:2298`	☐
The record is marked “valid us state code” `VALID-US-STATE-CODE`	otherwise mark as “input error”; mark as “state not ok”	Status	`COACTUPC.cbl:2495`	☐

`COCRDLIC` — List Credit Cards

When	Then	Type	Source	Owner confirms
Card account id equals account id `CARD-ACCT-ID = CC-ACCT-ID`	otherwise mark as “exclude this record”; jump to filter records exit	Validation	`COCRDLIC.cbl:1386`	☐
Card number equals card number n `CARD-NUM = CC-CARD-NUM-N`	otherwise mark as “exclude this record”; jump to filter records exit	Validation	`COCRDLIC.cbl:1397`	☐

`COPAUA0C` — Card Authorization Decision Program

When	Then	Type	Source	Owner confirms
Processed is more than reqsts process limit `WS-MSG-PROCESSED > WS-REQSTS-PROCESS-LIMIT`	mark as “loop end”	Money	`COPAUA0C.cbl:339`	☐
Transaction amount is more than available amount `WS-TRANSACTION-AMT > WS-AVAILABLE-AMT`	mark as “decline auth”; mark as “insufficient fund”	Money	`COPAUA0C.cbl:668`	☐

`CORPT00C` — Print Transaction reports by submitting batch

When	Then	Type	Source	Owner confirms
Curdate month is more than 12 `WS-CURDATE-MONTH > 12`	add 1 to curdate year; set curdate month to 1	Timing	`CORPT00C.cbl:225`	☐

`COSGN00C` — Signon Screen for the CardDemo Application

When	Then	Type	Source	Owner confirms
Usr pwd equals user pwd `SEC-USR-PWD = WS-USER-PWD`	set cdemo from tranid to tranid; set cdemo from program to pgmname	Validation	`COSGN00C.cbl:223`	☐

Risk register

Each risk with a suggested owner and how we would manage it.

ID	Risk	Impact	Where	How we manage it	Owner
R01	Unstructured jumps between sections of logic If not managed: some logic paths are missed or reordered in conversion.	High	CBSTM03A, CBSTM03B, COACTUPC, COCRDLIC, COCRDUPC and 2 more	Rebuild as structured logic and prove each path with regression tests generated from the old code.	Engineering
R02	Depends on a hierarchical IMS database If not managed: conversion is blocked until the data model is decided.	High	CBPAUP0C, COPAUA0C, COPAUS0C, COPAUS1C, DBUNLDGS and 2 more	Agree the target data model with your data architects before converting these programs.	Data architecture
R03	Many decision paths in one program If not managed: untested paths behave differently after conversion.	High	COACTUPC, COCRDLIC, COCRDUPC, COTRTLIC, COTRTUPC	Capture the current behaviour of every decision path as tests before any change.	QA
R04	Very large single program If not managed: long conversion and test cycle; edge cases slip through.	High	COACTUPC, COTRTLIC	Split by business function and convert and verify one slice at a time.	Engineering
R05	Program changes its own control flow at run time If not managed: hidden execution paths are translated incorrectly.	High	CBSTM03A	Restructure by hand with an expert, never by direct translation; test every path the old code could take.	Engineering lead with business SME
R06	Referenced code missing from what we received If not managed: scope and estimate are incomplete.	High	COCRDSEC	Send the missing source so it can be analysed and scoped.	Your application team
R07	Online errors not checked explicitly If not managed: failures surface differently to users after conversion.	Medium	COACTVWC, COADM01C, COCRDLIC, CODATE01, COMEN01C and 9 more	Define the intended error behaviour explicitly and test failure paths, not just success paths.	Engineering
R08	Money arithmetic on packed decimals If not managed: balances, interest or fees differ by rounding.	Medium	CBPAUP0C, CBSTM03A, COACCT01, COACTUPC, COBIL00C and 7 more	Use exact decimal types with identical precision and rounding, and prove it by comparing outputs on production-like data.	Finance SME and QA
R09	Writes data shared with many programs If not managed: data migration has to be coordinated across teams.	Medium	CBACT01C, CBACT04C, CBEXPORT, CBTRN02C, COACTUPC and 2 more	Migrate the programs that share this data together, or put one data service in front of it first.	Data team
R10	Years stored as two digits If not managed: dates are misread around the century window.	Medium	CBPAUP0C, DBUNLDGS, PAUDBLOD, PAUDBUNL	Identify the century rule in use and make it explicit in the new code, with tests on both sides of the boundary.	QA
R11	Exchanges messages over MQ If not managed: downstream systems receive changed message formats.	Medium	COACCT01, CODATE01, COPAUA0C	Freeze message formats as contracts and run a bridge so old and new systems can coexist.	Integration team

ID	Risk	Impact	Where	How we manage it	Owner
R12	Assembler module called from COBOL If not managed: behaviour has to be specified and rebuilt by hand.	Medium	`COBDATFT`, `MVSWAIT`	Specify its behaviour with your team and rebuild it as a small library function.	Engineering
R13	File errors not checked explicitly If not managed: bad input is processed silently or stops the job.	Medium	`CBSTM03A`	Define what should happen on bad or missing files and test it.	Engineering
R14	Code that is never reached If not managed: effort spent converting logic nobody uses.	Low	`COACTUPC`, `COCRDLIC`, `COCRDSLC`	Confirm with runtime coverage, then leave it out of the conversion.	Engineering
R15	Little documentation in the code If not managed: knowledge sits with a few people.	Low	`COACCT01`, `CODATE01`	Use the generated program profiles as a starting point and have SMEs confirm them early.	Your SMEs
R16	Program with no known caller If not managed: paying to convert something that is no longer used.	Low	`CBTRN01C`	Check scheduler and run history; retire it if it has not run recently.	Your operations team

What this assessment could not see

Nothing is silently skipped. These are the gaps in what we received or could confirm; each one is either closed by your team or carried as an explicit assumption.

329	205	22	13
files delivered	analysed	not analysed	open unknowns

ID	Type	What we could not see	Why it matters	How to close it
U-001	Missing source	Program COCRDSEC was referenced but not delivered `CDV1`	Anything it does is outside this assessment, and the programs that use it cannot be fully documented or tested.	Send COCRDSEC from the source library, or confirm it is a vendor or system module.
U-002	Inferred links	89 calls in 20 programs (COACTUPC, COACTVWC, COADM01C, COBIL00C and 16 more) go through a variable; the targets were traced from data, not written literally `COACTUPC → COACTUPC (app/cbl/COACTUPC.cbl:956)`, `COACTUPC → COMEN01C (app/cbl/COACTUPC.cbl:956)`, `COACTVWC → COACTVWC (app/cbl/COACTVWC.cbl:349)`, `COACTVWC → COMEN01C (app/cbl/COACTVWC.cbl:349)` and 85 more	These links are very likely right but have not been confirmed by a run.	Confirm with a developer, or with run-time statistics.
U-003	Not analysed: Assembler	Assembler module ASMWAIT was not analysed `app/maclib/ASMWAIT.mac`	Nothing in the delivered COBOL calls it; it may be unused.	Ask a developer to describe what it does, or confirm it is retired.
U-004	Not analysed: Assembler	Assembler module COBDATFT was not analysed `app/asm/COBDATFT.asm`, `CBACT01C`	Its behaviour is not documented here and must be specified by hand before conversion.	Ask a developer to describe what it does, or confirm it is retired.
U-005	Not analysed: Assembler	Assembler module COCDATFT was not analysed `app/maclib/COCDATFT.mac`	Nothing in the delivered COBOL calls it; it may be unused.	Ask a developer to describe what it does, or confirm it is retired.
U-006	Not analysed: Assembler	Assembler module MVSWAIT was not analysed `app/asm/MVSWAIT.asm`, `COBSWAIT`	Its behaviour is not documented here and must be specified by hand before conversion.	Ask a developer to describe what it does, or confirm it is retired.
U-007	Not analysed: CA-7 schedule	1 CA-7 schedule file delivered but not analysed `app/scheduler/CardDemo.ca7`	Job order, dependencies and timing are not on the map, so batch sequencing in the roadmap is based on JCL alone.	Review the schedule with your operations team when the waves are planned.
U-008	Not analysed: Control-M schedule	1 Control-M schedule file delivered but not analysed `app/scheduler/CardDemo.controlm`	Job order, dependencies and timing are not on the map, so batch sequencing in the roadmap is based on JCL alone.	Review the schedule with your operations team when the waves are planned.

ID	Type	What we could not see	Why it matters	How to close it
U-009	Not analysed: IMS DBD	4 IMS DBD files delivered but not analysed `app/app-authorization-ims-db2-mq/ims/DBPAUTP0.dbd`, `app/app-authorization-ims-db2-mq/ims/DBPAUTX0.dbd`, `app/app-authorization-ims-db2-mq/ims/PADFLDBD.DBD`, `app/app-authorization-ims-db2-mq/ims/PASFLDBD.DBD`	The structure of the IMS databases is not modelled, so their target data design is open.	Review with your data architect when the IMS programs are scoped.
U-010	Not analysed: IMS PSB	4 IMS PSB files delivered but not analysed `app/app-authorization-ims-db2-mq/ims/DLIGSAMP.PSB`, `app/app-authorization-ims-db2-mq/ims/PAUTBUNL.PSB`, `app/app-authorization-ims-db2-mq/ims/PSBPAUTB.psb`, `app/app-authorization-ims-db2-mq/ims/PSBPAUTL.psb`	Which IMS segments each program may access is not modelled.	Review with the IMS programs when they are scoped.
U-011	Not analysed: Utility control card	8 Utility control card files delivered but not analysed `app/app-transaction-type-db2/ctl/DB2CREAT.ctl`, `app/app-transaction-type-db2/ctl/DB2FREE.ctl`, `app/app-transaction-type-db2/ctl/DB2LTCAT.ctl`, `app/app-transaction-type-db2/ctl/DB2LTTYP.ctl` and 4 more	Sort and utility parameters can hold business logic such as record selection and field mapping.	Review the control cards for logic that must move with the jobs that use them.
U-012	Data from outside	26 datasets are read by the delivered jobs but created by nothing in the delivered code `AWS.M2.CARDDEMO.ACCTDATA.PS`, `AWS.M2.CARDDEMO.CARDDATA.PS`, `AWS.M2.CARDDEMO.CARDXREF.PS`, `AWS.M2.CARDDEMO.CUSTDATA.PS` and 4 more	They are fed by other systems, file transfers or manual steps; those feeds must keep working after modernization.	Ask operations where each one comes from (upstream system, transfer job, schedule).
U-013	Data leaving the estate	18 datasets are created but never read by the delivered code `AWS.CUSTDATA.CLUSTER`, `AWS.M2.CARDDEMO.ACCTDATA.ARRYPS`, `AWS.M2.CARDDEMO.ACCTDATA.IMPORT`, `AWS.M2.CARDDEMO.ACCTDATA.PSCOMP` and 4 more	Something downstream probably consumes them (reports, partners, other systems); the new system must keep producing them in the same format.	Confirm the consumers of each file and whether its layout is fixed.

Data stores and the batch catalog

Every dataset the application reads or writes, what kind it is, and which jobs create and use it. Files that come from or go to other systems must keep flowing after modernization.

73 datasets used by the application

18 VSAM files

10 generation data groups

26 fed from outside the delivered code

73 datasets in total, the 30 most used shown here; the full list with every job and step is in catalog.csv. 26 temporary datasets passed between steps are not listed. Built from JCL dispositions, IDCAMS definitions and CICS file definitions.

Dataset	Type	Created by	Read by
`AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS` CICS file TRANSACT	VSAM (KSDS)	`TRANBKP.STEP10`, `TRANFILE.STEP10`	`CBEXPORT.STEP02`, `COMBTRAN.STEP10`, `CREASTMT.STEP010`, `POSTTRAN.STEP15` and 4 more programs: CBEXPORT, CBTRN02C
`AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS` CICS file CCXREF	VSAM (KSDS)	`XREFFILE.STEP10`	`CBEXPORT.STEP02`, `CREASTMT.STEP040`, `INTCALC.STEP15`, `POSTTRAN.STEP15` and 3 more programs: CBACT03C, CBACT04C, CBEXPORT, CBSTM03A
`AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS` CICS file ACCTDAT	VSAM (KSDS)	`ACCTFILE.STEP10`	`ACCTFILE.STEP15`, `CBEXPORT.STEP02`, `CREASTMT.STEP040`, `INTCALC.STEP15` and 2 more programs: CBACT01C, CBACT04C, CBEXPORT, CBSTM03A
`AWS.M2.CARDDEMO.TRANSACT.BKUP`	GDG	`REPRTEST.STEP05`, `TRANBKP.STEP05R`, `TRANREPT.STEP05R`	`COMBTRAN.STEP05R`, `SORTTEST.STEP05R`, `TRANREPT.STEP05R`
`AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS` CICS file CUSTDAT	VSAM (KSDS)	`CUSTFILE.STEP10`	`CBEXPORT.STEP02`, `CREASTMT.STEP040`, `CUSTFILE.STEP15`, `READCUST.STEP05` programs: CBCUS01C, CBEXPORT, CBSTM03A
`AWS.M2.CARDDEMO.ESDSRRDS.PS`	VSAM	`ESDSRRDS.STEP01`	`ESDSRRDS.STEP03`, `ESDSRRDS.STEP05`
`AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS`	VSAM (KSDS)	`TCATBALF.STEP10`	`INTCALC.STEP15`, `POSTTRAN.STEP15`, `PRTCATBL.STEP05R`, `TCATBALF.STEP15` programs: CBACT04C, CBTRN02C
`AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS` CICS file CARDDAT	VSAM (KSDS)	`CARDFILE.STEP10`	`CARDFILE.STEP15`, `CBEXPORT.STEP02`, `READCARD.STEP05` programs: CBACT02C, CBEXPORT
`AWS.M2.CARDDEMO.PAUTDB.CHILD.FILEO`	SEQ	`UNLDPADB.STEP01`	`LOADPADB.STEP01` programs: PAUDBLOD, PAUDBUNL
`AWS.M2.CARDDEMO.PAUTDB.ROOT.FILEO`	SEQ	`UNLDPADB.STEP01`	`LOADPADB.STEP01` programs: PAUDBLOD, PAUDBUNL
`AWS.M2.CARDDEMO.TRANSACT.DALY`	GDG	`SORTTEST.STEP05R`, `TRANREPT.STEP05R`	`TRANREPT.STEP10R` programs: CBTRN03C

Dataset	Type	Created by	Read by
AWS.M2.CARDDEMO.USRSEC.PS	SEQ	DUSRSECJ.STEP01	DUSRSECJ.STEP03
OEM.IMS.IMSP.PAUTHDB Comes from outside the delivered code	SEQ	not created here	DBPAUTP0.UNLOAD, UNLDGSAM.STEP01, UNLDPADB.STEP01 programs: DBUNLDGS, PAUDBUNL
OEM.IMS.IMSP.PAUTHDBX Comes from outside the delivered code	SEQ	not created here	DBPAUTP0.UNLOAD, UNLDGSAM.STEP01, UNLDPADB.STEP01 programs: DBUNLDGS, PAUDBUNL
AWS.M2.CARDDEMO.ACCTDATA.ARRYPS Not read by the delivered code (goes elsewhere)	SEQ	READACCT.STEP05	programs: CBACT01C
AWS.M2.CARDDEMO.ACCTDATA.PSCOMP Not read by the delivered code (goes elsewhere)	SEQ	READACCT.STEP05	programs: CBACT01C
AWS.M2.CARDDEMO.ACCTDATA.VBPS Not read by the delivered code (goes elsewhere)	SEQ	READACCT.STEP05	programs: CBACT01C
AWS.M2.CARDDEMO.DISCGRP.PS Comes from outside the delivered code	SEQ	not created here	DEFGDGD.STEP60, DISCGRP.STEP15
AWS.M2.CARDDEMO.DISCGRP.VSAM.KSDS	VSAM (KSDS)	DISCGRP.STEP10	DISCGRP.STEP15, INTCALC.STEP15 programs: CBACT04C
AWS.M2.CARDDEMO.EXPORT.DATA	VSAM (KSDS)	CBEXPORT.STEP01	CBEXPORT.STEP02, CBIMPORT.STEP01 programs: CBEXPORT, CBIMPORT
AWS.M2.CARDDEMO.IMSDATA.DBPAUTP0 Comes from outside the delivered code	SEQ	not created here	DBPAUTP0.UNLOAD
AWS.M2.CARDDEMO.STATEMNT.PS Comes from outside the delivered code	SEQ	not created here	TXT2PDF1.TXT2PDF
AWS.M2.CARDDEMO.SYSTRAN	GDG	INTCALC.STEP15	COMBTRAN.STEP05R programs: CBACT04C
AWS.M2.CARDDEMO.TCATBALF.BKUP	GDG	PRTCATBL.STEP05R	PRTCATBL.STEP10R
AWS.M2.CARDDEMO.TCATBALF.REPT Not read by the delivered code (goes elsewhere)	SEQ	PRTCATBL.STEP10R	
AWS.M2.CARDDEMO.TRANCATG.PS Comes from outside the delivered code	SEQ	not created here	DEFGDGD.STEP40, TRANCATG.STEP15
AWS.M2.CARDDEMO.TRANCATG.VSAM.KSDS	VSAM (KSDS)	TRANCATG.STEP10	TRANCATG.STEP15, TRANREPT.STEP10R programs: CBTRN03C
AWS.M2.CARDDEMO.TRANSACT.COMBINED	GDG	COMBTRAN.STEP05R	COMBTRAN.STEP10
AWS.M2.CARDDEMO.TRANTYPE.PS Comes from outside the delivered code	SEQ	not created here	DEFGDGD.STEP20, TRANTYPE.STEP15

Dataset	Type	Created by	Read by
AWS.M2.CARDDEMO.TRANTYPE.VSAM.KSDS	VSAM (KSDS)	TRANTYPE.STEP10	TRANREPT.STEP10R, TRANTYPE.STEP15 programs: CBTRN03C

Field-level data lineage

How each important field moves through the programs, files, tables and screens, traced statement by statement from the code itself.

11,321
statements parsed, after expanding 290 copybook uses

12,994
data items in the symbol table

6,742
field-to-field data flows traced

32
dynamic calls resolved by tracing values

Every program was parsed the way the compiler sees it: copybooks expanded (with REPLACING applied, used 40 times), every statement read, and every MOVE, COMPUTE, file read and write, SQL host variable, screen field, CALL parameter and COMMAREA followed at the level of individual fields. This is deterministic code analysis; no AI is involved. The full graph is in lineage.json and can be queried with `modassess lineage`. Copybooks not delivered, so not expanded: CMQGMOV, CMQMDV, CMQODV, CMQPMOV, CMQTML, CMQV.

Where the money fields come from and go

Field	Changed by	Comes from	Ends up in
ACCT-CURR-BAL used in 13 programs	CBACT04C, CBTRN02C, COBIL00C	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.DALYTRAN.PS; AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS; AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS	AWS.M2.CARDDEMO.ACCTDATA.PSCOMP; AWS.M2.CARDDEMO.ACCTDATA.ARRYPS; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS
ACCT-CURR-CYC-CREDIT used in 13 programs	CBTRN02C	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.DALYTRAN.PS; COACTUP.CACTUPA; AWS.M2.CARDDEMO.EXPORT.DATA	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.ACCTDATA.PSCOMP; AWS.M2.CARDDEMO.EXPORT.DATA; COACTVW.CACTVWA
TRAN-AMT used in 9 programs	copied only, never calculated	AWS.M2.CARDDEMO.TRANSACT.DALY; AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.DALYTRAN.PS	AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA; AWS.M2.CARDDEMO.ACCTDATA.PSCOMP
ACCT-CURR-CYC-DEBIT used in 13 programs	CBTRN02C	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.DALYTRAN.PS; COACTUP.CACTUPA; AWS.M2.CARDDEMO.EXPORT.DATA	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA; COACTVW.CACTVWA; COACTUP.CACTUPA
ACCT-CASH-CREDIT-LIMIT used in 13 programs	copied only, never calculated	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; COACTUP.CACTUPA; AWS.M2.CARDDEMO.EXPORT.DATA	AWS.M2.CARDDEMO.ACCTDATA.PSCOMP; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA; COACTVW.CACTVWA
ACCT-CREDIT-LIMIT used in 13 programs	copied only, never calculated	AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; COACTUP.CACTUPA; AWS.M2.CARDDEMO.EXPORT.DATA	AWS.M2.CARDDEMO.ACCTDATA.PSCOMP; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA; COACTVW.CACTVWA
CUST-FICO-CREDIT-SCORE used in 8 programs	copied only, never calculated	AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA	AWS.M2.CARDDEMO.EXPORT.DATA; AWS.M2.CARDDEMO.CUSTDATA.IMPORT; CBSTM03A.STMTFILE; COACTVW.CACTVWA
DALYTRAN-AMT used in 2 programs	copied only, never calculated	AWS.M2.CARDDEMO.DALYTRAN.PS; CBTRN01C.DALYTRAN	AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS; AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS; AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS; AWS.M2.CARDDEMO.EXPORT.DATA

Field	Changed by	Comes from	Ends up in
`WS-TOTAL-INT` used in 1 program	`CBACT04C`	`AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS`; `AWS.M2.CARDDEMO.DISCGRP.VSAM.KSDS`; `AWS.M2.CARDDEMO.DALYTRAN.PS`	`AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS`; `AWS.M2.CARDDEMO.ACCTDATA.PSCOMP`; `AWS.M2.CARDDEMO.ACCTDATA.ARRYPS`; `AWS.M2.CARDDEMO.EXPORT.DATA`
`ACUP-NEW-CURR-BAL-N` used in 1 program	`COACTUPC`	`COACTUP.CACTUPA`	`AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS`; `COACTUP.CACTUPA`; `AWS.M2.CARDDEMO.ACCTDATA.PSCOMP`; `AWS.M2.CARDDEMO.ACCTDATA.ARRYPS`

Nearest data stores first (datasets, DB2 tables, screens, queues). Each step in the trace cites the statement that moves the data, for example `modassess lineage <run> ACCT-CURR-BAL --direction up`.

Program calls found by tracing values

These calls name the target program in a variable, so a text search cannot see them. The analysis followed the values assigned to the variable (VALUE clauses, MOVEs, dispatch tables). They are included in the dependency map at lower confidence and should be confirmed by your team.

Program	How	Variable	Possible targets
`COACTUPC` `COACTUPC.cbl:956`	CICS XCTL	`CDEMO-TO-PROGRAM`	COMEN01C
`COACTVWC` `COACTVWC.cbl:349`	CICS XCTL	`CDEMO-TO-PROGRAM`	COMEN01C
`COADM01C` `COADM01C.cbl:145`	CICS XCTL	`CDEMO-ADMIN-OPT-PGMNAME`	COTRTLIC, COTRTUPC, COUSR00C, COUSR01C, COUSR02C, COUSR03C
`COADM01C` `COADM01C.cbl:168`	CICS XCTL	`CDEMO-TO-PROGRAM`	COSGN00C
`COBIL00C` `COBIL00C.cbl:281`	CICS XCTL	`CDEMO-TO-PROGRAM`	COMEN01C, COSGN00C
`COCRDLIC` `COCRDLIC.cbl:402`	CICS XCTL	`LIT-MENUPGM`	COMEN01C
`COCRDLIC` `COCRDLIC.cbl:538`	CICS XCTL	`CCARD-NEXT-PROG`	COCRDSLC, COCRDUPC
`COCRDLIC` `COCRDLIC.cbl:566`	CICS XCTL	`CCARD-NEXT-PROG`	COCRDSLC, COCRDUPC
`COCRDSLC` `COCRDSLC.cbl:331`	CICS XCTL	`CDEMO-TO-PROGRAM`	COMEN01C
`COCRDUPC` `COCRDUPC.cbl:473`	CICS XCTL	`CDEMO-TO-PROGRAM`	COMEN01C
`COMEN01C` `COMEN01C.cbl:156`	CICS XCTL	`CDEMO-MENU-OPT-PGMNAME`	COACTUPC, COACTVWC, COBIL00C, COCRDLIC, COCRDSLC, COCRDUPC, COPAUS0C, CORPT00C and 3 more
`COMEN01C` `COMEN01C.cbl:184`	CICS XCTL	`CDEMO-MENU-OPT-PGMNAME`	COACTUPC, COACTVWC, COBIL00C, COCRDLIC, COCRDSLC, COCRDUPC, COPAUS0C, CORPT00C and 3 more

20 more in deep-analysis.json.

Code that can never run

Paragraphs that no PERFORM, GO TO or fall-through can reach. They need not be converted, but confirm first that nothing outside the program depends on them.

- `COCRDLIC`: SEND-LONG-TEXT

Option: rehost as is first

Rehosting moves the COBOL as it is onto a COBOL runtime on Linux or cloud (for example Rocket Enterprise Server, formerly Micro Focus) that emulates CICS, JCL and VSAM. It is usually the fastest, lowest-risk way off the mainframe and a common first step before selective modernization.

22
ready to rehost as is (14,527 lines)

20
ready with platform support (7,995 lines)

2
need work first (371 lines)

95%
of programs could move without code changes

Program	Status	What stands in the way
CBACT01C	Needs work	calls Assembler (COBDATFT), which must be rewritten or supported on the target; calls IBM Language Environment services
COBSWAIT	Needs work	calls Assembler (MVSWAIT), which must be rewritten or supported on the target
CBACT02C	Ready with platform support	calls IBM Language Environment services
CBACT03C	Ready with platform support	calls IBM Language Environment services
CBACT04C	Ready with platform support	calls IBM Language Environment services
CBCUS01C	Ready with platform support	calls IBM Language Environment services
CBEXPORT	Ready with platform support	calls IBM Language Environment services
CBIMPORT	Ready with platform support	calls IBM Language Environment services
CBPAUP0C	Ready with platform support	IMS DL/I needs IMS support on the target runtime
CBSTM03A	Ready with platform support	calls IBM Language Environment services; uses memory addresses (POINTER / ADDRESS OF); check on the target
CBTRN01C	Ready with platform support	calls IBM Language Environment services
CBTRN02C	Ready with platform support	calls IBM Language Environment services
CBTRN03C	Ready with platform support	calls IBM Language Environment services
COACCT01	Ready with platform support	IBM MQ needs an MQ client on the target
CODATE01	Ready with platform support	IBM MQ needs an MQ client on the target
COPAUA0C	Ready with platform support	IMS DL/I needs IMS support on the target runtime; IBM MQ needs an MQ client on the target; uses memory addresses (POINTER / ADDRESS OF); check on the target
COPAUS0C	Ready with platform support	IMS DL/I needs IMS support on the target runtime
COPAUS1C	Ready with platform support	IMS DL/I needs IMS support on the target runtime
CSUTLDTC	Ready with platform support	calls IBM Language Environment services
DBUNLDGS	Ready with platform support	IMS DL/I needs IMS support on the target runtime
PAUDBLOD	Ready with platform support	IMS DL/I needs IMS support on the target runtime

Program	Status	What stands in the way
PAUDBUNL	Ready with platform support	IMS DL/I needs IMS support on the target runtime

Error handling to preserve

How each program reacts when something fails is part of its behaviour. A converted program must fail the same way (same return codes, abends, messages and retries) or downstream jobs and operators will react differently.

15 programs check file I/O results (FILE STATUS, AT END, INVALID KEY)

23 programs handle CICS conditions (RESP, HANDLE CONDITION)

9 programs ignore some CICS errors on purpose (NOHANDLE)

4 programs check SQL results (SQLCODE, WHENEVER)

16 programs abend deliberately on serious errors

7 programs set a return code for the scheduler

9 programs commit or roll back work explicitly

6 programs hold record or row locks (READ UPDATE, ENQ, FOR UPDATE)

3 programs handle credentials or call security services

32 programs depend on the current date or time

Program	Error handling found	Not protected
`CBSTM03A`	It abends deliberately on serious errors. Writes 33 DISPLAY messages (operators may rely on them).	2 file operations with no FILE STATUS, AT END or INVALID KEY handling

43 other programs handle the errors we can detect. Each program's full error handling is in its profile.

Program profiles

What each program does, what it touches, and why we recommend what we do.

CBACT01C — READ THE ACCOUNT FILE AND WRITE INTO FILES

Convert

Source
app/cbl/CBACT01C.cbl · 358 lines

Roadmap stage
Batch foundations

Run by
READACCT

Calls
COBDATFT

Data
AWS.M2.CARDDEMO.ACCTDATA.ARRYPS, AWS.M2.CARDDEMO.ACCTDATA.PSCOMP, AWS.M2.CARDDEMO.ACCTDATA.VBPS, AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS

Platform
VSAM

Parsed
190 statements, 110 data items, 16 paragraphs

Rehost as is
Needs work
calls Assembler (COBDATFT), which must be rewritten or supported on the target
calls IBM Language Environment services

Evidence record
4288f4b6-9d6e-5057-b6fa-6937f06f26f2

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBACT01C is a batch program using VSAM. Its stated function is: READ THE ACCOUNT FILE AND WRITE INTO FILES CBACT01C.cbl:1.

How it works

It is started by batch job READACCT.

Main processing steps, in source order: acctfile get next CBACT01C.cbl:165; display acct record CBACT01C.cbl:200; popul acct record CBACT01C.cbl:215; write acct record CBACT01C.cbl:242; popul array record CBACT01C.cbl:253; write arry record CBACT01C.cbl:263; popul vbrc record CBACT01C.cbl:276; write vb1 record CBACT01C.cbl:287; and further steps.

It transfers control to or calls: COBDATFT (calls).

Inputs and outputs

- ACCTFILE (file): read CBACT01C.cbl:29
- OUTFILE (file): write CBACT01C.cbl:35
- ARRYFILE (file): write CBACT01C.cbl:40
- VBRCFILE (file): write CBACT01C.cbl:45

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 30 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (4 status fields declared): 11 CBACT01C.cbl:167
- Calls an abend routine (CEE3ABD): 1 CBACT01C.cbl:410

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.

CBACT02C — Read and print card data file

Convert

Source
`app/cbl/CBACT02C.cbl` · 129 lines

Roadmap stage
Batch foundations

Run by
READCARD

Data
AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS

Platform
VSAM

Parsed
63 statements, 28 data items, 5 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`c19d5551-f0e8-5dc9-b78f-87c8d55d8db8`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBACT02C is a batch program using VSAM. Its stated function is: Read and print card data file `CBACT02C.cbl:1`.

How it works

It is started by batch job `READCARD`.

Main processing steps, in source order: cardfile get next `CBACT02C.cbl:92`.

Inputs and outputs

- CARDFILE (file): read `CBACT02C.cbl:29`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 9 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (1 status field declared): 4 `CBACT02C.cbl:94`
- Calls an abend routine (CEE3ABD): 1 `CBACT02C.cbl:158`

Modernization notes

- No elevated modernization risk signals detected.

CBACT03C — Read and print account cross reference data file

Convert

Source
`app/cbl/CBACT03C.cbl` · 130 lines

Roadmap stage
Batch foundations

Run by
READXREF

Data
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS

Platform
VSAM

Parsed
64 statements, 25 data items, 5 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`b1243f84-fba4-5b6a-afb3-5258f37ba1f9`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBACT03C is a batch program using VSAM. Its stated function is: Read and print account cross reference data file `CBACT03C.cbl:1`.

How it works

It is started by batch job `READXREF`.

Main processing steps, in source order: xreffile get next `CBACT03C.cbl:92`.

Inputs and outputs

- XREFFILE (file): read `CBACT03C.cbl:29`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 10 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (1 status field declared): 4 `CBACT03C.cbl:94`
- Calls an abend routine (CEE3ABD): 1 `CBACT03C.cbl:158`

Modernization notes

- No elevated modernization risk signals detected.

CBACT04C

This is a interest calculator program

Convert

Source
`app/cbl/CBACT04C.cbl` · 552 lines

Roadmap stage
Batch foundations

Run by
INTCALC

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS,
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS,
AWS.M2.CARDDEMO.DISCGRP.VSAM.KSDS,
AWS.M2.CARDDEMO.SYSTRAN,
AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS

Platform
VSAM

Parsed
294 statements, 136 data items, 22 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`f4b57bec-7524-518d-8e4d-b4af9e79c658`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBACT04C is a batch program using VSAM. Its stated function is: This is a interest calculator program `CBACT04C.cbl:1`.

How it works

It is started by batch job `INTCALC`.

Main processing steps, in source order: tcatbalf get next `CBACT04C.cbl:325`; update account `CBACT04C.cbl:350`; get acct data `CBACT04C.cbl:372`; get xref data `CBACT04C.cbl:393`; get interest rate `CBACT04C.cbl:415`; a get default int rate `CBACT04C.cbl:443`; compute interest `CBACT04C.cbl:462`; b write tx `CBACT04C.cbl:473`; and further steps.

Inputs and outputs

- TCATBALF (file): read `CBACT04C.cbl:28`
- XREFFILE (file): read `CBACT04C.cbl:34`
- DISCGRP (file): read `CBACT04C.cbl:47`
- ACCTFILE (file): update `CBACT04C.cbl:41`
- TRANSACT (file): write `CBACT04C.cbl:53`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 27 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (5 status fields declared): 19 `CBACT04C.cbl:237`
- INVALID KEY handlers (record not found / duplicate key): 3 `CBACT04C.cbl:374`
- Calls an abend routine (CEE3ABD): 1 `CBACT04C.cbl:632`
- Depends on the current date or time `CBACT04C.cbl:614`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.

Business rules

- When interest rate is not 0, compute interest; compute fees `CBACT04C.cbl:214`

CBCUS01C — Read and print customer data file

Convert

Source
`app/cbl/CBCUS01C.cbl` · 130 lines

Roadmap stage
Batch foundations

Run by
READCUST

Data
AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS

Platform
VSAM

Parsed
64 statements, 40 data items, 5 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`da37d0a3-854e-5657-9c6b-15f1d1f5c0d3`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBCUS01C is a batch program using VSAM. Its stated function is: Read and print customer data file `CBCUS01C.cbl:1`.

How it works

It is started by batch job `READCUST`.

Main processing steps, in source order: custfile get next `CBCUS01C.cbl:92`.

Inputs and outputs

- CUSTFILE (file): read `CBCUS01C.cbl:29`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 10 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (1 status field declared): 4 `CBCUS01C.cbl:94`
- Calls an abend routine (CEE3ABD): 1 `CBCUS01C.cbl:158`

Modernization notes

- No elevated modernization risk signals detected.

CBEXPORT Export Customer Data for Branch Migration

Convert

Source
app/cbl/CBEXPORT.cbl · 396 lines

Roadmap stage
Batch foundations

Run by
CBEXPORT

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS, AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS, AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS, AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS, AWS.M2.CARDDEMO.EXPORT.DATA, AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
VSAM

Parsed
224 statements, 164 data items, 21 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
fd52520a-4366-535e-bb71-cb3719db8f74

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBEXPORT is a batch program using VSAM. Its stated function is: Export Customer Data for Branch Migration CBEXPORT.cbl:1.

How it works

It is started by batch job CBEXPORT.

Main processing steps, in source order: main processing CBEXPORT.cbl:149; initialize CBEXPORT.cbl:161; generate timestamp CBEXPORT.cbl:172; export customers CBEXPORT.cbl:243; read customer record CBEXPORT.cbl:258; create customer exp rec CBEXPORT.cbl:269; export accounts CBEXPORT.cbl:312; read account record CBEXPORT.cbl:327; and further steps.

Inputs and outputs

- CUSTFILE (file): read CBEXPORT.cbl:35
- ACCTFILE (file): read CBEXPORT.cbl:41
- XREFFILE (file): read CBEXPORT.cbl:47
- TRANSACT (file): read CBEXPORT.cbl:53
- CARDFILE (file): read CBEXPORT.cbl:59
- EXPFILE (file): write CBEXPORT.cbl:65

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 37 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (6 status fields declared): 0
- Calls an abend routine (CEE3ABD): 1 CBEXPORT.cbl:579
- Depends on the current date or time CBEXPORT.cbl:175

Modernization notes

- No elevated modernization risk signals detected.

CBIMPORT Import Customer Data from Branch Migration Export

Convert

Source
`app/cbl/CBIMPORT.cbl` · 337 lines

Roadmap stage
Batch foundations

Run by
CBIMPORT

Data
AWS.M2.CARDDEMO.ACCTDATA.IMPORT,
AWS.M2.CARDDEMO.CARDXREF.IMPORT,
AWS.M2.CARDDEMO.CUSTDATA.IMPORT,
AWS.M2.CARDDEMO.EXPORT.DATA,
AWS.M2.CARDDEMO.IMPORT.ERRORS,
AWS.M2.CARDDEMO.TRANSACT.IMPORT

Platform
VSAM

Parsed
181 statements, 165 data items, 16 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`54508bef-6d89-5472-ab75-e0271eb1e830`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBIMPORT is a batch program using VSAM. Its stated function is: Import Customer Data from Branch Migration Export `CBIMPORT.cbl:1`.

How it works

It is started by batch job `CBIMPORT`.

Main processing steps, in source order: main processing `CBIMPORT.cbl:165`; initialize `CBIMPORT.cbl:174`; process export file `CBIMPORT.cbl:248`; read export record `CBIMPORT.cbl:259`; process record by type `CBIMPORT.cbl:270`; process customer record `CBIMPORT.cbl:288`; process account record `CBIMPORT.cbl:323`; process xref record `CBIMPORT.cbl:352`; and further steps.

Inputs and outputs

- EXPFILE (file): read `CBIMPORT.cbl:37`
- CUSTOUT (file): write `CBIMPORT.cbl:43`
- ACCTOUT (file): write `CBIMPORT.cbl:48`
- XREFOUT (file): write `CBIMPORT.cbl:53`
- TRNXOUT (file): write `CBIMPORT.cbl:58`
- CARDOUT (file): write `CBIMPORT.cbl:63`
- ERROUT (file): write `CBIMPORT.cbl:68`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 29 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (7 status fields declared): 0
- Calls an abend routine (CEE3ABD): 1 `CBIMPORT.cbl:484`
- Depends on the current date or time `CBIMPORT.cbl:178`

Modernization notes

- No elevated modernization risk signals detected.

CBPAUP0C Delete Expired Pending Authoriation Messages

Convert with SME review

Source
`app/app-authorization-ims-db2-mq/cbl/CBPAUP0C.cbl` · 266 lines

Roadmap stage
Expert-reviewed core

Run by
CBPAUP0J

Platform
IMS

Parsed
117 statements, 86 data items, 17 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
`0ea8e6f7-1964-55bb-90bd-28fe19d48527`

Review
Pending SME review

Our recommendation: Convert with SME review

- Stores years as two digits and applies a century rule that must be made explicit.
- Uses IMS; the data model must be agreed with your data architects.

Purpose

CBPAUP0C is a batch program using IMS. Its stated function is: Delete Expired Pending Authoriation Messages `CBPAUP0C.cbl:1`.

How it works

It is started by batch job `CBPAUP0J`.

Main processing steps, in source order: main para `CBPAUP0C.cbl:136`; initialize `CBPAUP0C.cbl:183`; find next auth summary `CBPAUP0C.cbl:216`; find next auth dtl `CBPAUP0C.cbl:248`; check if expired `CBPAUP0C.cbl:277`; delete auth dtl `CBPAUP0C.cbl:303`; delete auth summary `CBPAUP0C.cbl:328`; take checkpoint `CBPAUP0C.cbl:352`.

Inputs and outputs

- No file, table or database access detected.

Error handling and operations

It sets a return code for the scheduler. Writes 29 DISPLAY messages (operators may rely on them).

- Sets RETURN-CODE (signals the batch scheduler): 1 `CBPAUP0C.cbl:382`
- Depends on the current date or time `CBPAUP0C.cbl:186`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

Business rules

- When approved auth count is at most 0 and approved auth count is at most 0, delete auth summary `CBPAUP0C.cbl:156`
- When diff is at least expiry days, mark as “qualified for delete” `CBPAUP0C.cbl:284`

CBSTM03A Print Account Statements from Transaction data

Convert with SME review

Source
app/cbl/CBSTM03A.CBL · 773 lines

Roadmap stage
Expert-reviewed core

Run by
CREASTMT

Calls
CBSTM03B

Parsed
440 statements, 182 data items, 25 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services
uses memory addresses (POINTER / ADDRESS OF); check on the target

Evidence record
8a702add-cd92-5058-ba70-0986b79d7569

Review
Pending SME review

Our recommendation: Convert with SME review

- Uses ALTER, which rewires its own control flow at run time; it must be restructured by hand.
- Performs money arithmetic on packed decimals; rounding must be proven identical.
- Jumps around with GO TO; its paths need confirming before translation.

Purpose

CBSTM03A is a batch program. Its stated function is: Print Account Statements from Transaction data CBSTM03A.CBL:1.

How it works

It is started by batch job CREASTMT.

Main processing steps, in source order: start CBSTM03A.CBL:296; mainline CBSTM03A.CBL:316; goback CBSTM03A.CBL:341; xreffile get next CBSTM03A.CBL:345; custfile get CBSTM03A.CBL:368; acctfile get CBSTM03A.CBL:392; trnxfile get CBSTM03A.CBL:416; create statement CBSTM03A.CBL:458; and further steps.

It transfers control to or calls: CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls), CBSTM03B (calls).

Inputs and outputs

- STMTFILE (file): write CBSTM03A.CBL:39
- HTMLFILE (file): write CBSTM03A.CBL:40

Error handling and operations

It abends deliberately on serious errors. Writes 33 DISPLAY messages (operators may rely on them).

- Calls an abend routine (CEE3ABD): 1 CBSTM03A.CBL:923
- Not protected: 2 file operations with no FILE STATUS, AT END or INVALID KEY handling.
- Uses memory addresses (POINTER / ADDRESS OF) CBSTM03A.CBL:266

Modernization notes

- 14 GO TO statements — control flow must be restructured, not transliterated.
- 4 ALTER statements (self-modifying flow) — highest translation risk.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.

Business rules

- When xref card number equals card num CBSTM03A.CBL:420

CBSTM03B

Does file processing related to Transact Report

Convert with SME review

Source
app/cbl/CBSTM03B.CBL · 162 lines

Roadmap stage
Expert-reviewed core

Called by
CBSTM03A

Platform
VSAM

Parsed
58 statements, 33 data items, 14 paragraphs

Rehost as is
Ready

Evidence record
895082de-0bcf-54b2-bb66-a8abdde0617e

Review
Pending SME review

Our recommendation: Convert with SME review

- Jumps around with GO TO; its paths need confirming before translation.

Purpose

CBSTM03B is a called subprogram using VSAM. Its stated function is: Does file processing related to Transact Report CBSTM03B.CBL:1.

How it works

It is invoked by CBSTM03A.

Main processing steps, in source order: start CBSTM03B.CBL:116; goback CBSTM03B.CBL:130; trnxfile proc CBSTM03B.CBL:133; xreffile proc CBSTM03B.CBL:157; custfile proc CBSTM03B.CBL:181; acctfile proc CBSTM03B.CBL:206.

Inputs and outputs

- TRNXFILE (file): read CBSTM03B.CBL:31
- XREFFILE (file): read CBSTM03B.CBL:37
- CUSTFILE (file): read CBSTM03B.CBL:43
- ACCTFILE (file): read CBSTM03B.CBL:49

Error handling and operations

It checks file I/O results.

- FILE STATUS checks (4 status fields declared): 0

Modernization notes

- 13 GO TO statements — control flow must be restructured, not transliterated.

CBTRN01C — Post the records from daily transaction file

Confirm and retire

Source
`app/cbl/CBTRN01C.cbl` · 415 lines

Roadmap stage
Before scoping

Platform
VSAM

Parsed
216 statements, 130 data items, 18 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`1d171f2b-d5c4-5a44-87b2-0f341bf3f000`

Review
Pending SME review

Our recommendation: Confirm and retire

- No program, batch job or online transaction invokes it in the supplied source.

Purpose

CBTRN01C is a batch program using VSAM. Its stated function is: Post the records from daily transaction file `CBTRN01C.cbl:1`.

How it works

No caller, job or transaction in the analysed estate invokes it (possible dead code or invoked from outside the supplied source).

Main processing steps, in source order: main para `CBTRN01C.cbl:155`; dalytran get next `CBTRN01C.cbl:202`; lookup xref `CBTRN01C.cbl:227`; read account `CBTRN01C.cbl:241`.

Inputs and outputs

- DALYTRAN (file): read `CBTRN01C.cbl:29`
- CUSTFILE (file): read `CBTRN01C.cbl:34`
- XREFFILE (file): read `CBTRN01C.cbl:40`
- CARDFILE (file): read `CBTRN01C.cbl:46`
- ACCTFILE (file): read `CBTRN01C.cbl:52`
- TRANFILE (file): read `CBTRN01C.cbl:58`

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 28 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (6 status fields declared): 14 `CBTRN01C.cbl:204`
- INVALID KEY handlers (record not found / duplicate key): 2 `CBTRN01C.cbl:231`
- Calls an abend routine (CEE3ABD): 1 `CBTRN01C.cbl:473`

Modernization notes

- No elevated modernization risk signals detected.

CBTRN02C

Post the records from daily transaction file

Convert

Source
app/cbl/CBTRN02C.cbl · 619 lines

Roadmap stage
Batch foundations

Run by
POSTTRAN

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS,
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS,
AWS.M2.CARDDEMO.DALYREJS,
AWS.M2.CARDDEMO.DALYTRAN.PS,
AWS.M2.CARDDEMO.TCATBALF.VSAM.KSDS,
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
VSAM

Parsed
339 statements, 146 data items, 26 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
a014b529-9c60-5f8b-8711-aed39acb8536

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBTRN02C is a batch program using VSAM. Its stated function is: Post the records from daily transaction file CBTRN02C.cbl:1.

How it works

It is started by batch job POSTTRAN.

Main processing steps, in source order: dalytran get next CBTRN02C.cbl:345; validate tran CBTRN02C.cbl:370; a lookup xref CBTRN02C.cbl:380; b lookup acct CBTRN02C.cbl:393; post transaction CBTRN02C.cbl:424; write reject rec CBTRN02C.cbl:446; update tcatbal CBTRN02C.cbl:467; a create tcatbal rec CBTRN02C.cbl:503; and further steps.

Inputs and outputs

- DALYTRAN (file): read CBTRN02C.cbl:29
- TRANFILE (file): write CBTRN02C.cbl:34
- XREFFILE (file): read CBTRN02C.cbl:40
- DALYREJS (file): write CBTRN02C.cbl:46
- ACCTFILE (file): update CBTRN02C.cbl:51
- TCATBALF (file): update CBTRN02C.cbl:57

Error handling and operations

It checks file I/O results, abends deliberately on serious errors and sets a return code for the scheduler. Writes 26 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (6 status fields declared): 19 CBTRN02C.cbl:239
- INVALID KEY handlers (record not found / duplicate key): 4 CBTRN02C.cbl:384
- Sets RETURN-CODE (signals the batch scheduler): 1 CBTRN02C.cbl:230
- Calls an abend routine (CEE3ABD): 1 CBTRN02C.cbl:711
- Depends on the current date or time CBTRN02C.cbl:693

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.

Business rules

- When account credit limit is at least temp balance, otherwise set validation fail reason to 102 CBTRN02C.cbl:407
- When account expiration date is at least dalytran original ts, otherwise set validation fail reason to 103 CBTRN02C.cbl:414
- When dalytran amount is at least 0, add dalytran amount to account current cyc credit CBTRN02C.cbl:548

CBTRN03C Print the transaction detail report Convert

Source
app/cbl/CBTRN03C.cbl · 545 lines

Roadmap stage
Pilot

Run by
TRANREPT

Data
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS, AWS.M2.CARDDEMO.DATEPARM, AWS.M2.CARDDEMO.TRANCATG.VSAM.KSDS, AWS.M2.CARDDEMO.TRANREPT, AWS.M2.CARDDEMO.TRANSACT.DALY, AWS.M2.CARDDEMO.TRANTYPE.VSAM.KSDS

Platform
VSAM

Parsed
320 statements, 137 data items, 26 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
4d2055ee-6e34-50ce-a414-11e4f7fa3a08

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

CBTRN03C is a batch program using VSAM. Its stated function is: Print the transaction detail report CBTRN03C.cbl:1.

How it works

It is started by batch job TRANREPT.

Main processing steps, in source order: dateparm read CBTRN03C.cbl:220; tranfile get next CBTRN03C.cbl:248; write transaction report CBTRN03C.cbl:274; write page totals CBTRN03C.cbl:293; write account totals CBTRN03C.cbl:306; write grand totals CBTRN03C.cbl:318; write headers CBTRN03C.cbl:324; write report rec CBTRN03C.cbl:343; and further steps.

Inputs and outputs

- TRANFILE (file): read CBTRN03C.cbl:29
- TRANREPT (file): write CBTRN03C.cbl:51
- CARDXREF (file): read CBTRN03C.cbl:33
- TRANTYPE (file): read CBTRN03C.cbl:39
- TRANCATG (file): read CBTRN03C.cbl:45
- DATEPARM (file): read CBTRN03C.cbl:55

Error handling and operations

It checks file I/O results and abends deliberately on serious errors. Writes 27 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (6 status fields declared): 15 CBTRN03C.cbl:222
- INVALID KEY handlers (record not found / duplicate key): 3 CBTRN03C.cbl:486
- Calls an abend routine (CEE3ABD): 1 CBTRN03C.cbl:630

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.

Business rules

- When current card number is not transaction card number CBTRN03C.cbl:181

COACCT01 Online program

Convert with SME review

Source
`app/app-vsam-mq/cbl/COACCT01.cbl` · 500 lines

Roadmap stage
Expert-reviewed core

Run by
CDRA

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS

Platform
CICS, MQ

Parsed
224 statements, 109 data items, 13 paragraphs

Rehost as is
Ready with platform support
IBM MQ needs an MQ client on the target

Evidence record
`16978ad9-4d1f-5c48-bb52-7b1f195e803b`

Review
Pending SME review

Our recommendation: Convert with SME review

- Performs money arithmetic on packed decimals; rounding must be proven identical.

Purpose

COACCT01 is an online (CICS) program using CICS, MQ. The source has no header describing its purpose; intent below is derived from its structure and data access `COACCT01.cbl:1`.

How it works

It is started by CICS transaction `CDRA`.

Main processing steps, in source order: control `COACCT01.cbl:178`; main process `COACCT01.cbl:325`; get request `COACCT01.cbl:334`; process request reply `COACCT01.cbl:390`; put reply `COACCT01.cbl:462`; error `COACCT01.cbl:501`; termination `COACCT01.cbl:538`.

Inputs and outputs

- ACCTDAT (cics_file): read `COACCT01.cbl:396`

Error handling and operations

It handles CICS conditions. Writes 2 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 2 `COACCT01.cbl:191`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COACCT01.cbl:326`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- MQ messaging — preserve message contracts and ordering semantics.

COACTUPC — Accept and process ACCOUNT UPDATE

Wrap and defer

Source
app/cbl/COACTUPC.cbl · 3,368 lines

Roadmap stage
Wrap and decide later

Run by
CAUP

Called by
COMEN01C

Calls
COMEN01C

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS, AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH, AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS

Platform
CICS

Parsed
1,570 statements, 1,194 data items, 101 paragraphs

Rehost as is
Ready

Evidence record
addcc8aa-a5a9-5129-b2ee-010ee3d4cff7

Review
Pending SME review

Our recommendation: Wrap and defer

- 3 jobs, screens or programs depend on it.
- Safer to keep it running behind an interface while the programs around it move first.

Purpose

COACTUPC is an online (CICS) program using CICS. Its stated function is: Accept and process ACCOUNT UPDATE COACTUPC.cbl:1.

How it works

It is started by CICS transaction CAUP.

It is invoked by COMEN01C.

Main processing steps, in source order: main COACTUPC.cbl:859; common return COACTUPC.cbl:1007; process inputs COACTUPC.cbl:1025; receive map COACTUPC.cbl:1039; edit map inputs COACTUPC.cbl:1429; compare old new COACTUPC.cbl:1681; edit account COACTUPC.cbl:1783; edit mandatory COACTUPC.cbl:1824; and further steps.

It transfers control to or calls: COMEN01C (CICS XCTL).

Inputs and outputs

- CXACAIX (cics_file): read COACTUPC.cbl:3654
- ACCTDAT (cics_file): read, update COACTUPC.cbl:3703
- CUSTDAT (cics_file): read, update COACTUPC.cbl:3753
- Screen map COACTUP (3270 terminal I/O)

Error handling and operations

It handles CICS conditions and abends deliberately on serious errors.

- CICS commands that capture a response code (RESP): 9 COACTUPC.cbl:1040
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 2 COACTUPC.cbl:862
- CICS commands with NOHANDLE (errors deliberately ignored): 1 COACTUPC.cbl:4211
- Deliberate abends (EXEC CICS ABEND): 1 COACTUPC.cbl:4222
- Commits work explicitly (EXEC CICS SYNCPOINT) COACTUPC.cbl:952
- Rolls back on failure (SYNCPOINT ROLLBACK / SQL ROLLBACK) COACTUPC.cbl:4099
- Reads records for update (record locks held until rewrite or commit) COACTUPC.cbl:3894
- Depends on the current date or time COACTUPC.cbl:2671

Modernization notes

- 51 GO TO statements — control flow must be restructured, not transliterated.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- High decision complexity (cyclomatic ≈438) — split by function before converting.

Business rules

- When the record is marked “valid general purp code”, otherwise mark as “input error”; mark as “edit us phonea not ok” COACTUPC.cbl:2298
- When the record is marked “valid us state code”, otherwise mark as “input error”; mark as “state not ok” COACTUPC.cbl:2495

COACTVWC — Accept and process Account View request

Convert

Source
app/cbl/COACTVWC.cbl · 703 lines

Roadmap stage
Online screens and services

Run by
CAVW

Called by
COMEN01C

Calls
COMEN01C

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS,
AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH,
AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS

Platform
CICS

Parsed
319 statements, 657 data items, 37 paragraphs

Rehost as is
Ready

Evidence record
91982f66-70a7-5479-85a7-bc2340c9c91c

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COACTVWC is an online (CICS) program using CICS. Its stated function is: Accept and process Account View request COACTVWC.cbl:1.

How it works

It is started by CICS transaction CAVW.

It is invoked by COMEN01C.

Main processing steps, in source order: main COACTVWC.cbl:262; common return COACTVWC.cbl:394; send map COACTVWC.cbl:416; screen init COACTVWC.cbl:431; setup screen vars COACTVWC.cbl:460; setup screen attrs COACTVWC.cbl:541; send screen COACTVWC.cbl:577; process inputs COACTVWC.cbl:596; and further steps.

It transfers control to or calls: COMEN01C (CICS XCTL).

Inputs and outputs

- CXACAIX (cics_file): read COACTVWC.cbl:727
- ACCTDAT (cics_file): read COACTVWC.cbl:776
- CUSTDAT (cics_file): read COACTVWC.cbl:826
- Screen map COACTVW (3270 terminal I/O)

Error handling and operations

It handles CICS conditions and abends deliberately on serious errors.

- CICS commands that capture a response code (RESP): 5 COACTVWC.cbl:583
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 2 COACTVWC.cbl:264
- CICS commands with NOHANDLE (errors deliberately ignored): 1 COACTVWC.cbl:924
- Deliberate abends (EXEC CICS ABEND): 1 COACTVWC.cbl:934
- Depends on the current date or time COACTVWC.cbl:434

Modernization notes

- 9 GO TO statements — control flow must be restructured, not transliterated.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COADM01C Admin Menu for Admin users

Convert

Source
app/cbl/COADM01C.cbl · 189 lines

Roadmap stage
Online screens and services

Run by
CA00

Called by
COSGN00C, COTRTLIC, COTRTUPC, COUSR00C, COUSR01C, COUSR02C, COUSR03C

Calls
COSGN00C, COTRTLIC, COTRTUPC, COUSR00C, COUSR01C, COUSR02C, COUSR03C

Platform
CICS

Parsed
94 statements, 359 data items, 8 paragraphs

Rehost as is
Ready

Evidence record
b8cdc4a3-eafe-542c-81c2-f9d1d5cb5102

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COADM01C is an online (CICS) program using CICS. Its stated function is: Admin Menu for Admin users COADM01C.cbl:1.

How it works

It is started by CICS transaction CA00.

It is invoked by COSGN00C, COTRTLIC, COTRTUPC, COUSR00C, COUSR01C, COUSR02C, COUSR03C.

Main processing steps, in source order: main para COADM01C.cbl:75; process enter key COADM01C.cbl:119; return to signon screen COADM01C.cbl:163; send menu screen COADM01C.cbl:175; receive menu screen COADM01C.cbl:192; populate header info COADM01C.cbl:205; build menu options COADM01C.cbl:229; pgmiderr err para COADM01C.cbl:270.

It transfers control to or calls: COSGN00C (CICS XCTL), COTRTLIC (traced via CDEMO-ADMIN-OPT-PGMNAME), COTRTUPC (traced via CDEMO-ADMIN-OPT-PGMNAME), COUSR00C (traced via CDEMO-ADMIN-OPT-PGMNAME), COUSR01C (traced via CDEMO-ADMIN-OPT-PGMNAME), COUSR02C (traced via CDEMO-ADMIN-OPT-PGMNAME), COUSR03C (traced via CDEMO-ADMIN-OPT-PGMNAME).

Inputs and outputs

- No file, table or database access detected.
- Screen map COADM01 (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 1 COADM01C.cbl:194
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 1 COADM01C.cbl:77
- Depends on the current date or time COADM01C.cbl:207

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COBIL00C Bill Payment - Pay account balance in full and a

Convert

Source
app/cbl/COBIL00C.cbl · 420 lines

Roadmap stage
Online screens and services

Run by
CB00

Called by
COMEN01C

Calls
COMEN01C, COSGN00C

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS, AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH, AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
CICS

Parsed
222 statements, 254 data items, 16 paragraphs

Rehost as is
Ready

Evidence record
07970bec-c330-5307-bfe2-0660b0ab74b6

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COBIL00C is an online (CICS) program using CICS. Its stated function is: Bill Payment - Pay account balance in full and a `COBIL00C.cbl:1`.

How it works

It is started by CICS transaction `CB00`.

It is invoked by `COMEN01C`.

Main processing steps, in source order: main para `COBIL00C.cbl:99`; process enter key `COBIL00C.cbl:154`; get current timestamp `COBIL00C.cbl:249`; return to prev screen `COBIL00C.cbl:273`; send billpay screen `COBIL00C.cbl:289`; receive billpay screen `COBIL00C.cbl:306`; populate header info `COBIL00C.cbl:319`; read acctdat file `COBIL00C.cbl:343`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- ACCTDAT (cics_file): read, update `COBIL00C.cbl:345`
- CXACAIX (cics_file): read `COBIL00C.cbl:410`
- TRANSACT (cics_file): read, write `COBIL00C.cbl:443`
- Screen map `COBIL00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 6 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 7 `COBIL00C.cbl:308`
- Reads records for update (record locks held until rewrite or commit) `COBIL00C.cbl:345`
- Depends on the current date or time `COBIL00C.cbl:251`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COBSWAIT — UTILITY PROGRAM TO WAIT (PARM IN CENTISECONDS)

Convert

Source
`app/cbl/COBSWAIT.cbl` · 13 lines

Roadmap stage
Batch foundations

Run by
WAITSTEP

Calls
MVSWAIT

Parsed
4 statements, 2 data items, 0 paragraphs

Rehost as is
Needs work
calls Assembler (MVSWAIT), which must be rewritten or supported on the target

Evidence record
`cf24d2ec-5a8e-582a-9717-8274cb5f2886`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

COBSWAIT is a batch program. Its stated function is: UTILITY PROGRAM TO WAIT (PARM IN CENTISECONDS) `COBSWAIT.cbl:1`.

How it works

It is started by batch job `WAITSTEP`.

It transfers control to or calls: `MVSWAIT` (calls).

Inputs and outputs

- No file, table or database access detected.

Error handling and operations

No explicit error handling found; failures fall through to the platform's default behaviour.

Modernization notes

- No elevated modernization risk signals detected.

COBTUPDT — Update Transaction type based on user input

Convert

Source
app/app-transaction-type-db2/cbl/COBTUPDT.cbl · 177 lines

Roadmap stage
Batch foundations

Run by
MNTTRDB2

Data
TRANSACTION_TYPE, INPFILE

Platform
DB2

Parsed
71 statements, 22 data items, 9 paragraphs

Rehost as is
Ready

Evidence record
0ac04ead-43d6-51b5-8fea-c0df2dec2c73

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.
- Batch, so old and new versions can be run on the same input and their outputs compared.

Purpose

COBTUPDT is a batch program using DB2. Its stated function is: Update Transaction type based on user input `COBTUPDT.cbl:1`.

How it works

It is started by batch job `MNTTRDB2`.

Main processing steps, in source order: read next records `COBTUPDT.cbl:91`; read records `COBTUPDT.cbl:100`; treat record `COBTUPDT.cbl:109`; insert db `COBTUPDT.cbl:132`; update db `COBTUPDT.cbl:166`; delete db `COBTUPDT.cbl:196`.

Inputs and outputs

- TRANSACTION_TYPE (sql): delete, update, write `COBTUPDT.cbl:137`
- INPFILE (file): read `COBTUPDT.cbl:31`

Error handling and operations

It checks file I/O results, checks SQL results and sets a return code for the scheduler. Writes 11 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (1 status field declared): 1 `COBTUPDT.cbl:84`
- AT END handlers (end of file): 1 `COBTUPDT.cbl:102`
- SQLCODE / SQLSTATE checks: 8 `COBTUPDT.cbl:152`
- Sets RETURN-CODE (signals the batch scheduler): 1 `COBTUPDT.cbl:232`

Modernization notes

- Embedded DB2 SQL — usually portable; check SQLCODE handling and isolation levels.

COCRDLIC List Credit Cards

Convert with SME review

Source
`app/cbl/COCRDLIC.cbl` · 1,093 lines

Roadmap stage
Expert-reviewed core

Run by
CCLI

Called by
COMEN01C

Calls
COCRDSLC, COCRDUPC, COMEN01C

Data
AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS

Platform
CICS

Parsed
571 statements, 732 data items, 41 paragraphs; 1 unreachable (SEND-LONG-TEXT)

Rehost as is
Ready

Evidence record
`43910410-8255-5046-a59b-83a526fb33a2`

Review
Pending SME review

Our recommendation: Convert with SME review

- Jumps around with GO TO; its paths need confirming before translation.
- Contains many decision paths; an expert should confirm the edge cases.

Purpose

COCRDLIC is an online (CICS) program using CICS. Its stated function is: List Credit Cards `COCRDLIC.cbl:1`.

How it works

It is started by CICS transaction `CCLI`.

It is invoked by `COMEN01C`.

Main processing steps, in source order: main `COCRDLIC.cbl:298`; common return `COCRDLIC.cbl:604`; send map `COCRDLIC.cbl:624`; screen init `COCRDLIC.cbl:642`; screen array init `COCRDLIC.cbl:678`; setup array attribs `COCRDLIC.cbl:748`; setup screen attrs `COCRDLIC.cbl:837`; setup message `COCRDLIC.cbl:895`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL), `COCRDSLC` (CICS XCTL), `COCRDUPC` (CICS XCTL), `COCRDSLC` (CICS XCTL), `COCRDUPC` (CICS XCTL).

Inputs and outputs

- CARDDAT (cics_file): read `COCRDLIC.cbl:1129`
- Screen map `COCRDLI` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 8 `COCRDLIC.cbl:939`
- Depends on the current date or time `COCRDLIC.cbl:645`

Modernization notes

- 16 GO TO statements — control flow must be restructured, not transliterated.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

Business rules

- When card account id equals account id, otherwise mark as “exclude this record”; jump to filter records exit `COCRDLIC.cbl:1386`
- When card number equals card number n, otherwise mark as “exclude this record”; jump to filter records exit `COCRDLIC.cbl:1397`

COCRDSLC — Accept and process credit card detail request

Convert

Source
app/cbl/COCRDSLC.cbl · 642 lines

Roadmap stage
Online screens and services

Run by
CCDL

Called by
COCRDLIC, COMEN01C

Calls
COMEN01C

Data
AWS.M2.CARDDEMO.CARDDATA.VSAM.AIX.PATH, AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS

Platform
CICS

Parsed
307 statements, 372 data items, 36 paragraphs

Rehost as is
Ready

Evidence record
0b9e3acf-3e79-5c91-8af2-e76943b5fdff

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COCRDSLC is an online (CICS) program using CICS. Its stated function is: Accept and process credit card detail request COCRDSLC.cbl:1.

How it works

It is started by CICS transaction CCDL.

It is invoked by COCRDLIC, COMEN01C.

Main processing steps, in source order: main COCRDSLC.cbl:248; common return COCRDSLC.cbl:394; send map COCRDSLC.cbl:412; screen init COCRDSLC.cbl:427; setup screen vars COCRDSLC.cbl:457; setup screen attrs COCRDSLC.cbl:502; send screen COCRDSLC.cbl:563; process inputs COCRDSLC.cbl:582; and further steps.

It transfers control to or calls: COMEN01C (CICS XCTL).

Inputs and outputs

- CARDDAT (cics_file): read COCRDSLC.cbl:742
- CARDAIX (cics_file): read COCRDSLC.cbl:783
- Screen map COCRDSL (3270 terminal I/O)

Error handling and operations

It handles CICS conditions and abends deliberately on serious errors.

- CICS commands that capture a response code (RESP): 4 COCRDSLC.cbl:569
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 2 COCRDSLC.cbl:250
- CICS commands with NOHANDLE (errors deliberately ignored): 1 COCRDSLC.cbl:865
- Deliberate abends (EXEC CICS ABEND): 1 COCRDSLC.cbl:875
- Depends on the current date or time COCRDSLC.cbl:430

Modernization notes

- 9 GO TO statements — control flow must be restructured, not transliterated.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COCRDUPC — Accept and process credit card detail request

Convert with SME review

Source
`app/cbl/COCRDUPC.cbl` · 1,194 lines

Roadmap stage
Expert-reviewed core

Run by
CCUP

Called by
COCRDLIC, COMEN01C

Calls
COMEN01C

Data
AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS

Platform
CICS

Parsed
575 statements, 444 data items, 47 paragraphs

Rehost as is
Ready

Evidence record
`bd87ebd4-b693-5831-a7ac-acb320bad729`

Review
Pending SME review

Our recommendation: Convert with SME review

- Jumps around with GO TO; its paths need confirming before translation.
- Contains many decision paths; an expert should confirm the edge cases.

Purpose

COCRDUPC is an online (CICS) program using CICS. Its stated function is: Accept and process credit card detail request `COCRDUPC.cbl:1`.

How it works

It is started by CICS transaction `CCUP`.

It is invoked by `COCRDLIC`, `COMEN01C`.

Main processing steps, in source order: main `COCRDUPC.cbl:367`; common return `COCRDUPC.cbl:546`; process inputs `COCRDUPC.cbl:564`; receive map `COCRDUPC.cbl:578`; edit map inputs `COCRDUPC.cbl:641`; edit account `COCRDUPC.cbl:721`; edit card `COCRDUPC.cbl:762`; edit name `COCRDUPC.cbl:806`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL).

Inputs and outputs

- CARDDAT (cics_file): read, update `COCRDUPC.cbl:1382`
- Screen map `COCRDUP` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions and abends deliberately on serious errors.

- CICS commands that capture a response code (RESP): 5 `COCRDUPC.cbl:579`
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 2 `COCRDUPC.cbl:370`
- CICS commands with NOHANDLE (errors deliberately ignored): 1 `COCRDUPC.cbl:1539`
- Deliberate abends (EXEC CICS ABEND): 1 `COCRDUPC.cbl:1550`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COCRDUPC.cbl:469`
- Reads records for update (record locks held until rewrite or commit) `COCRDUPC.cbl:1427`
- Depends on the current date or time `COCRDUPC.cbl:1055`

Modernization notes

- 21 GO TO statements — control flow must be restructured, not transliterated.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- High decision complexity (cyclomatic ≈180) — split by function before converting.

CODATE01 Online program

Convert with SME review

Source
app/app-vsam-mq/cbl/CODATE01.cbl · 409 lines

Roadmap stage
Expert-reviewed core

Run by
CDRD

Platform
CICS, MQ

Parsed
198 statements, 66 data items, 13 paragraphs

Rehost as is
Ready with platform support
IBM MQ needs an MQ client on the target

Evidence record
047f3b87-cc9a-5ffe-99e8-f23642c645e2

Review
Pending SME review

Our recommendation: Convert with SME review

- Performs money arithmetic on packed decimals; rounding must be proven identical.

Purpose

CODATE01 is an online (CICS) program using CICS, MQ. The source has no header describing its purpose; intent below is derived from its structure and data access CODATE01.cbl:1.

How it works

It is started by CICS transaction CDRD.

Main processing steps, in source order: control CODATE01.cbl:127; main process CODATE01.cbl:274; get request CODATE01.cbl:283; process request reply CODATE01.cbl:339; put reply CODATE01.cbl:366; error CODATE01.cbl:405; termination CODATE01.cbl:442.

Inputs and outputs

- No file, table or database access detected.

Error handling and operations

It handles CICS conditions. Writes 2 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 1 CODATE01.cbl:140
- Commits work explicitly (EXEC CICS SYNCPOINT) CODATE01.cbl:275
- Depends on the current date or time CODATE01.cbl:343

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- MQ messaging — preserve message contracts and ordering semantics.

COMEN01C Main Menu for the Regular users

Convert

Source
app/cbl/COMEN01C.cbl · 213 lines

Roadmap stage
Online screens and services

Run by
CM00

Called by
COACTUPC, COACTVWC, COBIL00C, COCRDLIC, COCRDSLC, COCRDUPC, COPAUS0C, CORPT00C, COSGN00C, COTRN00C, COTRN01C, COTRN02C

Calls
COACTUPC, COACTVWC, COBIL00C, COCRDLIC, COCRDSLC, COCRDUPC, COPAUS0C, CORPT00C, COSGN00C, COTRN00C, COTRN01C, COTRN02C

Platform
CICS

Parsed
110 statements, 386 data items, 7 paragraphs

Rehost as is
Ready

Evidence record
0d90b54c-633f-5baf-9c93-e828b5ebaddb

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COMEN01C is an online (CICS) program using CICS. Its stated function is: Main Menu for the Regular users COMEN01C.cbl:1.

How it works

It is started by CICS transaction CM00.

It is invoked by COACTUPC, COACTVWC, COBIL00C, COCRDLIC, COCRDSLC, COCRDUPC, COPAUS0C, CORPT00C, COSGN00C, COTRN00C, COTRN01C, COTRN02C.

Main processing steps, in source order: main para COMEN01C.cbl:75; process enter key COMEN01C.cbl:115; return to signon screen COMEN01C.cbl:196; send menu screen COMEN01C.cbl:208; receive menu screen COMEN01C.cbl:225; populate header info COMEN01C.cbl:238; build menu options COMEN01C.cbl:262.

It transfers control to or calls: COSGN00C (CICS XCTL), COACTUPC (traced via CDEMO-MENU-OPT-PGMNAME), COACTVWC (traced via CDEMO-MENU-OPT-PGMNAME), COBIL00C (traced via CDEMO-MENU-OPT-PGMNAME), COCRDLIC (traced via CDEMO-MENU-OPT-PGMNAME), COCRDSLC (traced via CDEMO-MENU-OPT-PGMNAME), COCRDUPC (traced via CDEMO-MENU-OPT-PGMNAME), COPAUS0C (traced via CDEMO-MENU-OPT-PGMNAME), CORPT00C (traced via CDEMO-MENU-OPT-PGMNAME), COTRN00C (traced via CDEMO-MENU-OPT-PGMNAME), COTRN01C (traced via CDEMO-MENU-OPT-PGMNAME), COTRN02C (traced via CDEMO-MENU-OPT-PGMNAME).

Inputs and outputs

- No file, table or database access detected.
- Screen map COMEN01 (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 1 COMEN01C.cbl:227
- CICS commands with NOHANDLE (errors deliberately ignored): 1 COMEN01C.cbl:148
- Depends on the current date or time COMEN01C.cbl:240

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COPAUA0C Card Authorization Decision Program

Wrap and defer

Source
app/app-authorization-ims-db2-mq/cbl/COPAUA0C.cbl · 771 lines

Roadmap stage
Wrap and decide later

Run by
CP00

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS,
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS,
AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS

Platform
CICS, IMS, MQ

Parsed
365 statements, 198 data items, 43 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime
IBM MQ needs an MQ client on the target
uses memory addresses (POINTER / ADDRESS OF); check on the target

Evidence record
9d0f308b-5581-5c38-9e35-38b004810a3b

Review
Pending SME review

Our recommendation: Wrap and defer

- 1 job, screen or program depends on it.
- Reads a hierarchical IMS database whose target data model is not yet decided.
- Safer to keep it running behind an interface while the programs around it move first.

Purpose

COPAUA0C is an online (CICS) program using CICS, IMS, MQ. Its stated function is: Card Authorization Decision Program `COPAUA0C.cbl:1`.

How it works

It is started by CICS transaction `CP00`.

Main processing steps, in source order: main para `COPAUA0C.cbl:220`; initialize `COPAUA0C.cbl:230`; schedule psb `COPAUA0C.cbl:292`; main process `COPAUA0C.cbl:323`; extract request msg `COPAUA0C.cbl:351`; read request mq `COPAUA0C.cbl:386`; process auth `COPAUA0C.cbl:438`; read xref record `COPAUA0C.cbl:472`; and further steps.

Inputs and outputs

- CCXREF (cics_file): read `COPAUA0C.cbl:477`
- ACCTDAT (cics_file): read `COPAUA0C.cbl:525`
- CUSTDAT (cics_file): read `COPAUA0C.cbl:573`

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 3 `COPAUA0C.cbl:477`
- CICS commands with NOHANDLE (errors deliberately ignored): 4 `COPAUA0C.cbl:233`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COPAUA0C.cbl:334`
- Writes to a journal or transient-data queue (often an audit trail) `COPAUA0C.cbl:1001`
- Depends on the current date or time `COPAUA0C.cbl:857`
- Uses memory addresses (POINTER / ADDRESS OF) `COPAUA0C.cbl:730`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.
- MQ messaging — preserve message contracts and ordering semantics.

Business rules

- When processed is more than reqsts process limit, mark as “loop end” `COPAUA0C.cbl:339`
- When transaction amount is more than available amount, mark as “decline auth”; mark as “insufficient fund” `COPAUA0C.cbl:668`

COPAUS0C — Summary View of Authoriation Messages

Convert with SME review

Source
app/app-authorization-ims-db2-mq/cbl/COPAUS0C.cbl · 792 lines

Roadmap stage
Expert-reviewed core

Run by
CPVS

Called by
COMEN01C, COPAUS1C

Calls
COMEN01C, COPAUS1C, COSGN00C

Data
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS, AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH, AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS

Platform
CICS, IMS

Parsed
416 statements, 969 data items, 20 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
3141f906-9421-5c3e-ad8f-9d09b0511ee8

Review
Pending SME review

Our recommendation: Convert with SME review

- Performs money arithmetic on packed decimals; rounding must be proven identical.
- Uses IMS; the data model must be agreed with your data architects.

Purpose

COPAUS0C is an online (CICS) program using CICS, IMS. Its stated function is: Summary View of Authoriation Messages `COPAUS0C.cbl:1`.

How it works

It is started by CICS transaction `CPVS`.

It is invoked by `COMEN01C`, `COPAUS1C`.

Main processing steps, in source order: main para `COPAUS0C.cbl:178`; process enter key `COPAUS0C.cbl:261`; gather details `COPAUS0C.cbl:342`; process pf7 key `COPAUS0C.cbl:362`; process pf8 key `COPAUS0C.cbl:388`; process page forward `COPAUS0C.cbl:415`; get authorizations `COPAUS0C.cbl:458`; reposition authorizations `COPAUS0C.cbl:488`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL), `COPAUS1C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COMEN01C` (CICS XCTL), `COPAUS1C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- CXACAIX (cics_file): read `COPAUS0C.cbl:818`
- ACCTDAT (cics_file): read `COPAUS0C.cbl:869`
- CUSTDAT (cics_file): read `COPAUS0C.cbl:920`
- Screen map `COPAU00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 4 `COPAUS0C.cbl:715`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COPAUS0C.cbl:686`
- Depends on the current date or time `COPAUS0C.cbl:729`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

COPAUS1C — Detail View of Authorization Message

Convert with SME review

Source
`app/app-authorization-ims-db2-mq/cbl/COPAUS1C.cbl` · 461 lines

Roadmap stage
Expert-reviewed core

Run by
CPVD

Called by
COPAUS0C

Calls
COPAUS0C, COPAUS2C

Platform
CICS, IMS

Parsed
213 statements, 501 data items, 15 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
`efcf85a6-a393-53e9-8605-79bba4e028fe`

Review
Pending SME review

Our recommendation: Convert with SME review

- Uses IMS; the data model must be agreed with your data architects.

Purpose

COPAUS1C is an online (CICS) program using CICS, IMS. Its stated function is: Detail View of Authorization Message `COPAUS1C.cbl:1`.

How it works

It is started by CICS transaction `CPVD`.

It is invoked by `COPAUS0C`.

Main processing steps, in source order: main para `COPAUS1C.cbl:157`; process enter key `COPAUS1C.cbl:208`; mark auth fraud `COPAUS1C.cbl:230`; process pf8 key `COPAUS1C.cbl:268`; populate auth details `COPAUS1C.cbl:291`; return to prev screen `COPAUS1C.cbl:360`; send authview screen `COPAUS1C.cbl:373`; receive authview screen `COPAUS1C.cbl:398`; and further steps.

It transfers control to or calls: `COPAUS2C` (CICS LINK), `COPAUS0C` (CICS XCTL).

Inputs and outputs

- No file, table or database access detected.
- Screen map `COPAU01` (3270 terminal I/O)

Error handling and operations

It checks file I/O results. Writes 1 DISPLAY message (operators may rely on them).

- AT END handlers (end of file): 1 `COPAUS1C.cbl:320`
- CICS commands with NOHANDLE (errors deliberately ignored): 2 `COPAUS1C.cbl:248`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COPAUS1C.cbl:558`
- Rolls back on failure (SYNCPOINT ROLLBACK / SQL ROLLBACK) `COPAUS1C.cbl:566`
- Depends on the current date or time `COPAUS1C.cbl:411`

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

COPAUS2C Mark Authorization Message Fraud

Convert

Source
app/app-authorization-ims-db2-mq/cbl/COPAUS2C.cbl · 201 lines

Roadmap stage
Online screens and services

Called by
COPAUS1C

Data
AUTHFRDS

Platform
CICS, DB2

Parsed
56 statements, 61 data items, 2 paragraphs

Rehost as is
Ready

Evidence record
b1ca11ac-7ef5-5980-97a6-ef8ed335c2d1

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COPAUS2C is an online (CICS) program using CICS, DB2. Its stated function is: Mark Authorization Message Fraud `COPAUS2C.cbl:1`.

How it works

It is invoked by `COPAUS1C`.

Main processing steps, in source order: main para `COPAUS2C.cbl:89`; fraud update `COPAUS2C.cbl:221`.

Inputs and outputs

- AUTHFRDS (sql): update, write `COPAUS2C.cbl:141`

Error handling and operations

It checks SQL results.

- CICS commands with NOHANDLE (errors deliberately ignored): 2 `COPAUS2C.cbl:91`
- SQLCODE / SQLSTATE checks: 3 `COPAUS2C.cbl:199`
- Depends on the current date or time `COPAUS2C.cbl:91`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- Embedded DB2 SQL — usually portable; check SQLCODE handling and isolation levels.

CORPT00C

Print Transaction reports by submitting batch

Convert

Source
`app/cbl/CORPT00C.cbl` · 498 lines

Roadmap stage
Online screens and services

Run by
CR00

Called by
COMEN01C

Calls
COMEN01C, COSGN00C, CSUTLDTC

Platform
CICS

Parsed
239 statements, 364 data items, 10 paragraphs

Rehost as is
Ready

Evidence record
`6bd1e9ff-8475-50e3-aee2-9ce39638b0ee`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

CORPT00C is an online (CICS) program using CICS. Its stated function is: Print Transaction reports by submitting batch `CORPT00C.cbl:1`.

How it works

It is started by CICS transaction `CR00`.

It is invoked by `COMEN01C`.

Main processing steps, in source order: main para `CORPT00C.cbl:163`; process enter key `CORPT00C.cbl:208`; submit job to intrdr `CORPT00C.cbl:462`; wirte jobsub tdq `CORPT00C.cbl:515`; return to prev screen `CORPT00C.cbl:540`; send trnrpt screen `CORPT00C.cbl:556`; return to cics `CORPT00C.cbl:585`; receive trnrpt screen `CORPT00C.cbl:596`; and further steps.

It transfers control to or calls: `CSUTLDTC` (calls), `CSUTLDTC` (calls), `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- No file, table or database access detected.
- Screen map `CORPT00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 2 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 2 `CORPT00C.cbl:517`
- Writes to a journal or transient-data queue (often an audit trail) `CORPT00C.cbl:517`
- Depends on the current date or time `CORPT00C.cbl:215`

Modernization notes

- 1 GO TO statements — control flow must be restructured, not transliterated.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

Business rules

- When curdate month is more than 12, add 1 to curdate year; set curdate month to 1 `CORPT00C.cbl:225`

COSGN00C Signon Screen for the CardDemo Application

Convert

Source
`app/cbl/COSGN00C.cbl` · 172 lines

Roadmap stage
Online screens and services

Run by
CC00

Called by
COADM01C, COBIL00C, COMEN01C, COPAUS0C, CORPT00C, COTRN00C, COTRN01C, COTRN02C, COUSR00C, COUSR01C, COUSR02C, COUSR03C

Calls
COADM01C, COMEN01C

Data
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS

Platform
CICS

Parsed
80 statements, 223 data items, 6 paragraphs

Rehost as is
Ready

Evidence record
`2ad48207-4662-599b-bab4-c5086326354b`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COSGN00C is an online (CICS) program using CICS. Its stated function is: Signon Screen for the CardDemo Application `COSGN00C.cbl:1`.

How it works

It is started by CICS transaction `CC00`.

It is invoked by `COADM01C`, `COBIL00C`, `COMEN01C`, `COPAUS0C`, `CORPT00C`, `COTRN00C`, `COTRN01C`, `COTRN02C`, `COUSR00C`, `COUSR01C`, `COUSR02C`, `COUSR03C`.

Main processing steps, in source order: main para `COSGN00C.cbl:73`; process enter key `COSGN00C.cbl:108`; send signon screen `COSGN00C.cbl:145`; send plain text `COSGN00C.cbl:162`; populate header info `COSGN00C.cbl:177`; read user sec file `COSGN00C.cbl:209`.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COMEN01C` (CICS XCTL).

Inputs and outputs

- USRSEC (cics_file): read `COSGN00C.cbl:211`
- Screen map `COSGN00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 2 `COSGN00C.cbl:110`
- Depends on the current date or time `COSGN00C.cbl:179`
- Handles credential fields (PASSWDI, PASSWDL, SEC-USR-PWD)

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

Business rules

- When usr pwd equals user pwd, set cdemo from tranid to tranid; set cdemo from program to pgmname `COSGN00C.cbl:223`

COTRN00C List Transactions from TRANSACT file

Convert

Source
app/cbl/COTRN00C.cbl · 529 lines

Roadmap stage
Online screens and services

Run by
CT00

Called by
COMEN01C, COTRN01C

Calls
COMEN01C, COSGN00C, COTRN01C

Data
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
CICS

Parsed
344 statements, 819 data items, 16 paragraphs

Rehost as is
Ready

Evidence record
48b7c879-626d-5279-a6fa-39fe97c5bc07

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COTRN00C is an online (CICS) program using CICS. Its stated function is: List Transactions from TRANSACT file `COTRN00C.cbl:1`.

How it works

It is started by CICS transaction `CT00`.

It is invoked by `COMEN01C`, `COTRN01C`.

Main processing steps, in source order: main para `COTRN00C.cbl:95`; process enter key `COTRN00C.cbl:146`; process pf7 key `COTRN00C.cbl:234`; process pf8 key `COTRN00C.cbl:257`; process page forward `COTRN00C.cbl:279`; process page backward `COTRN00C.cbl:333`; populate tran data `COTRN00C.cbl:381`; initialize tran data `COTRN00C.cbl:450`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COTRN01C` (CICS XCTL), `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COTRN01C` (CICS XCTL).

Inputs and outputs

- TRANSACT (cics_file): read `COTRN00C.cbl:593`
- Screen map `COTRN00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 3 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 4 `COTRN00C.cbl:556`
- Depends on the current date or time `COTRN00C.cbl:569`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COTRN01C View a Transaction from TRANSACT file

Convert

Source
app/cbl/COTRN01C.cbl · 231 lines

Roadmap stage
Online screens and services

Run by
CT01

Called by
COMEN01C, COTRN00C

Calls
COMEN01C, COSGN00C, COTRN00C

Data
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
CICS

Parsed
105 statements, 359 data items, 9 paragraphs

Rehost as is
Ready

Evidence record
899b4235-93ad-5ce2-9b51-733a8c0718c2

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COTRN01C is an online (CICS) program using CICS. Its stated function is: View a Transaction from TRANSACT file `COTRN01C.cbl:1`.

How it works

It is started by CICS transaction `CT01`.

It is invoked by `COMEN01C`, `COTRN00C`.

Main processing steps, in source order: main para `COTRN01C.cbl:86`; process enter key `COTRN01C.cbl:144`; return to prev screen `COTRN01C.cbl:197`; send trnview screen `COTRN01C.cbl:213`; receive trnview screen `COTRN01C.cbl:230`; populate header info `COTRN01C.cbl:243`; read transact file `COTRN01C.cbl:267`; clear current screen `COTRN01C.cbl:301`; and further steps.

It transfers control to or calls: `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COTRN00C` (CICS XCTL).

Inputs and outputs

- TRANSACT (cics_file): read `COTRN01C.cbl:269`
- Screen map `COTRN01` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 1 DISPLAY message (operators may rely on them).

- CICS commands that capture a response code (RESP): 2 `COTRN01C.cbl:232`
- Reads records for update (record locks held until rewrite or commit) `COTRN01C.cbl:269`
- Depends on the current date or time `COTRN01C.cbl:245`

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COTRN02C — Add a new Transaction to TRANSACT file

Convert

Source
app/cbl/COTRN02C.cbl · 614 lines

Roadmap stage
Online screens and services

Run by
CT02

Called by
COMEN01C

Calls
COMEN01C, COSGN00C, CSUTLDTC

Data
AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH, AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS, AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS

Platform
CICS

Parsed
363 statements, 395 data items, 18 paragraphs

Rehost as is
Ready

Evidence record
4b09ed14-2af2-5596-8bb6-d6963dd59066

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COTRN02C is an online (CICS) program using CICS. Its stated function is: Add a new Transaction to TRANSACT file `COTRN02C.cbl:1`.

How it works

It is started by CICS transaction `CT02`.

It is invoked by `COMEN01C`.

Main processing steps, in source order: main para `COTRN02C.cbl:107`; process enter key `COTRN02C.cbl:164`; validate input key fields `COTRN02C.cbl:193`; validate input data fields `COTRN02C.cbl:235`; add transaction `COTRN02C.cbl:442`; copy last tran data `COTRN02C.cbl:471`; return to prev screen `COTRN02C.cbl:500`; send trnadd screen `COTRN02C.cbl:516`; and further steps.

It transfers control to or calls: `CSUTLDTC` (calls), `CSUTLDTC` (calls), `COMEN01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- CXACAIX (cics_file): read `COTRN02C.cbl:578`
- CCXREF (cics_file): read `COTRN02C.cbl:611`
- TRANSACT (cics_file): read, write `COTRN02C.cbl:644`
- Screen map `COTRN02` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 5 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 6 `COTRN02C.cbl:541`
- Depends on the current date or time `COTRN02C.cbl:554`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COTRTLIC List Transaction Type for updates and deletes

Convert with SME review

Source
app/app-transaction-type-db2/cbl/COTRTLIC.cbl · 1,597 lines

Roadmap stage
Expert-reviewed core

Run by
CTLI

Called by
COADM01C

Calls
COADM01C, COTRTUPC, DSNTIAC

Data
TRANSACTION_TYPE

Platform
CICS, DB2

Parsed
725 statements, 782 data items, 63 paragraphs

Rehost as is
Ready

Evidence record
be46146b-cea2-5c28-b3ce-2d25c26f694e

Review
Pending SME review

Our recommendation: Convert with SME review

- Performs money arithmetic on packed decimals; rounding must be proven identical.
- Jumps around with GO TO; its paths need confirming before translation.
- Contains many decision paths; an expert should confirm the edge cases.

Purpose

COTRTLIC is an online (CICS) program using CICS, DB2. Its stated function is: List Transaction Type for updates and deletes `COTRTLIC.cbl:1`.

How it works

It is started by CICS transaction `CTLI`.

It is invoked by `COADM01C`.

Main processing steps, in source order: main `COTRTLIC.cbl:498`; common return `COTRTLIC.cbl:899`; receive map `COTRTLIC.cbl:919`; receive screen `COTRTLIC.cbl:930`; edit inputs `COTRTLIC.cbl:960`; edit array `COTRTLIC.cbl:982`; edit array desc `COTRTLIC.cbl:1060`; edit typecd `COTRTLIC.cbl:1096`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COTRTUPC` (CICS XCTL), `COTRTUPC` (CICS XCTL), `DSNTIAC` (traced via LIT-DSNTIAC).

Inputs and outputs

- TRANSACTION_TYPE (sql): delete, read, update `COTRTLIC.cbl:338`
- Screen map `COTRTLI` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions and checks SQL results.

- CICS commands that capture a response code (RESP): 2 `COTRTLIC.cbl:931`
- SQLCODE / SQLSTATE checks: 16 `COTRTLIC.cbl:1635`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COTRTLIC.cbl:616`
- Depends on the current date or time `COTRTLIC.cbl:1296`

Modernization notes

- 28 GO TO statements — control flow must be restructured, not transliterated.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- Embedded DB2 SQL — usually portable; check SQLCODE handling and isolation levels.
- High decision complexity (cyclomatic ≈231) — split by function before converting.

COTRTUPC — Accept and process TRANSACTION TYPE UPDATE

Convert with SME review

Source
`app/app-transaction-type-db2/cbl/COTRTUPC.cbl` · 1,241 lines

Roadmap stage
Expert-reviewed core

Run by
CTTU

Called by
COADM01C, COTRTLIC

Calls
COADM01C

Data
TRANSACTION_TYPE

Platform
CICS, DB2

Parsed
568 statements, 409 data items, 65 paragraphs

Rehost as is
Ready

Evidence record
`dadc50f2-47b4-5a1d-8444-100435f75361`

Review
Pending SME review

Our recommendation: Convert with SME review

- Performs money arithmetic on packed decimals; rounding must be proven identical.
- Jumps around with GO TO; its paths need confirming before translation.
- Contains many decision paths; an expert should confirm the edge cases.

Purpose

COTRTUPC is an online (CICS) program using CICS, DB2. Its stated function is: Accept and process TRANSACTION TYPE UPDATE `COTRTUPC.cbl:1`.

How it works

It is started by CICS transaction `CTTU`.

It is invoked by `COADM01C`, `COTRTLIC`.

Main processing steps, in source order: main `COTRTUPC.cbl:345`; common return `COTRTUPC.cbl:559`; check pfkeys `COTRTUPC.cbl:577`; process inputs `COTRTUPC.cbl:625`; receive map `COTRTUPC.cbl:641`; store map in new `COTRTUPC.cbl:652`; edit map inputs `COTRTUPC.cbl:689`; compare old new `COTRTUPC.cbl:783`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL).

Inputs and outputs

- TRANSACTION_TYPE (sql): delete, read, update, write `COTRTUPC.cbl:1475`
- Screen map `COTRTUP` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions, checks SQL results and abends deliberately on serious errors.

- CICS commands that capture a response code (RESP): 2 `COTRTUPC.cbl:642`
- CICS HANDLE CONDITION / HANDLE ABEND / IGNORE CONDITION: 2 `COTRTUPC.cbl:348`
- CICS commands with NOHANDLE (errors deliberately ignored): 1 `COTRTUPC.cbl:1684`
- SQLCODE / SQLSTATE checks: 10 `COTRTUPC.cbl:1487`
- Deliberate abends (EXEC CICS ABEND): 1 `COTRTUPC.cbl:1695`
- Commits work explicitly (EXEC CICS SYNCPOINT) `COTRTUPC.cbl:453`
- Depends on the current date or time `COTRTUPC.cbl:1113`

Modernization notes

- 23 GO TO statements — control flow must be restructured, not transliterated.
- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.
- Embedded DB2 SQL — usually portable; check SQLCODE handling and isolation levels.
- High decision complexity (cyclomatic ≈218) — split by function before converting.

COUSR00C — List all users from USRSEC file

Convert

Source
`app/cbl/COUSR00C.cbl` · 531 lines

Roadmap stage
Online screens and services

Run by
CU00

Called by
COADM01C

Calls
COADM01C, COSGN00C, COUSR02C, COUSR03C

Data
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS

Platform
CICS

Parsed
340 statements, 818 data items, 16 paragraphs

Rehost as is
Ready

Evidence record
`c60547cc-321e-5dea-8b25-0b88a900831f`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COUSR00C is an online (CICS) program using CICS. Its stated function is: List all users from USRSEC file `COUSR00C.cbl:1`.

How it works

It is started by CICS transaction `CU00`.

It is invoked by `COADM01C`.

Main processing steps, in source order: main para `COUSR00C.cbl:98`; process enter key `COUSR00C.cbl:149`; process pf7 key `COUSR00C.cbl:237`; process pf8 key `COUSR00C.cbl:260`; process page forward `COUSR00C.cbl:282`; process page backward `COUSR00C.cbl:336`; populate user data `COUSR00C.cbl:384`; initialize user data `COUSR00C.cbl:446`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COUSR02C` (CICS XCTL), `COUSR03C` (CICS XCTL), `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COUSR02C` (CICS XCTL), `COUSR03C` (CICS XCTL), `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL), `COUSR02C` (CICS XCTL), `COUSR03C` (CICS XCTL).

Inputs and outputs

- USRSEC (cics_file): read `COUSR00C.cbl:588`
- Screen map `COUSR00` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 3 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 4 `COUSR00C.cbl:551`
- Depends on the current date or time `COUSR00C.cbl:564`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COUSR01C — Add a new Regular/Admin user to USRSEC file

Convert

Source
app/cbl/COUSR01C.cbl · 198 lines

Roadmap stage
Online screens and services

Run by
CU01

Called by
COADM01C

Calls
COADM01C, COSGN00C

Data
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS

Platform
CICS

Parsed
108 statements, 233 data items, 9 paragraphs

Rehost as is
Ready

Evidence record
25b80f2e-6e3b-58dc-ac27-521f3380f7ea

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COUSR01C is an online (CICS) program using CICS. Its stated function is: Add a new Regular/Admin user to USRSEC file `COUSR01C.cbl:1`.

How it works

It is started by CICS transaction `CU01`.

It is invoked by `COADM01C`.

Main processing steps, in source order: main para `COUSR01C.cbl:71`; process enter key `COUSR01C.cbl:115`; return to prev screen `COUSR01C.cbl:165`; send usradd screen `COUSR01C.cbl:184`; receive usradd screen `COUSR01C.cbl:201`; populate header info `COUSR01C.cbl:214`; write user sec file `COUSR01C.cbl:238`; clear current screen `COUSR01C.cbl:279`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- USRSEC (cics_file): write `COUSR01C.cbl:240`
- Screen map `COUSR01` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions.

- CICS commands that capture a response code (RESP): 2 `COUSR01C.cbl:203`
- Depends on the current date or time `COUSR01C.cbl:216`
- Handles credential fields (PASSWDI, PASSWDL, SEC-USR-PWD)

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COUSR02C — Update a user in USRSEC file

Convert

Source
app/cbl/COUSR02C.cbl · 303 lines

Roadmap stage
Online screens and services

Run by
CU02

Called by
COADM01C, COUSR00C

Calls
COADM01C, COSGN00C

Data
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS

Platform
CICS

Parsed
168 statements, 241 data items, 11 paragraphs

Rehost as is
Ready

Evidence record
8687cadd-40bc-54c1-8558-e374e02186d6

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COUSR02C is an online (CICS) program using CICS. Its stated function is: Update a user in USRSEC file `COUSR02C.cbl:1`.

How it works

It is started by CICS transaction `CU02`.

It is invoked by `COADM01C`, `COUSR00C`.

Main processing steps, in source order: main para `COUSR02C.cbl:82`; process enter key `COUSR02C.cbl:143`; update user info `COUSR02C.cbl:177`; return to prev screen `COUSR02C.cbl:250`; send usrupd screen `COUSR02C.cbl:266`; receive usrupd screen `COUSR02C.cbl:283`; populate header info `COUSR02C.cbl:296`; read user sec file `COUSR02C.cbl:320`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- USRSEC (cics_file): read, update `COUSR02C.cbl:322`
- Screen map `COUSR02` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 2 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 3 `COUSR02C.cbl:285`
- Reads records for update (record locks held until rewrite or commit) `COUSR02C.cbl:322`
- Depends on the current date or time `COUSR02C.cbl:298`
- Handles credential fields (PASSWDI, PASSWDL, SEC-USR-PWD)

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

COUSR03C — Delete a user from USRSEC file

Convert

Source
app/cbl/COUSR03C.cbl · 251 lines

Roadmap stage
Online screens and services

Run by
CU03

Called by
COADM01C, COUSR00C

Calls
COADM01C, COSGN00C

Data
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS

Platform
CICS

Parsed
131 statements, 229 data items, 11 paragraphs

Rehost as is
Ready

Evidence record
a73a520a-b09c-52b9-8606-b57334426eb4

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

COUSR03C is an online (CICS) program using CICS. Its stated function is: Delete a user from USRSEC file `COUSR03C.cbl:1`.

How it works

It is started by CICS transaction `CU03`.

It is invoked by `COADM01C`, `COUSR00C`.

Main processing steps, in source order: main para `COUSR03C.cbl:82`; process enter key `COUSR03C.cbl:142`; delete user info `COUSR03C.cbl:174`; return to prev screen `COUSR03C.cbl:197`; send usrdel screen `COUSR03C.cbl:213`; receive usrdel screen `COUSR03C.cbl:230`; populate header info `COUSR03C.cbl:243`; read user sec file `COUSR03C.cbl:267`; and further steps.

It transfers control to or calls: `COADM01C` (CICS XCTL), `COSGN00C` (CICS XCTL).

Inputs and outputs

- USRSEC (cics_file): delete, read `COUSR03C.cbl:269`
- Screen map `COUSR03` (3270 terminal I/O)

Error handling and operations

It handles CICS conditions. Writes 2 DISPLAY messages (operators may rely on them).

- CICS commands that capture a response code (RESP): 3 `COUSR03C.cbl:232`
- Reads records for update (record locks held until rewrite or commit) `COUSR03C.cbl:269`
- Depends on the current date or time `COUSR03C.cbl:245`

Modernization notes

- CICS pseudo-conversational screen logic — maps to a request/response service plus UI.

CSUTLDTC Subprogram program Convert

Source
`app/cbl/CSUTLDTC.cbl` · 114 lines

Roadmap stage
Batch foundations

Called by
CORPT00C, COTRN02C

Parsed
37 statements, 39 data items, 2 paragraphs

Rehost as is
Ready with platform support
calls IBM Language Environment services

Evidence record
`6661bd9b-4b7e-5ff8-abe3-ac69a58cf3a4`

Review
Pending SME review

Our recommendation: Convert

- Self-contained and conventional; suitable for AI translation with automated verification.

Purpose

CSUTLDTC is a called subprogram. The source has no header describing its purpose; intent below is derived from its structure and data access `CSUTLDTC.cbl:1`.

How it works

It is invoked by `CORPT00C`, `COTRN02C`.

Main processing steps, in source order: a000 main `CSUTLDTC.cbl:103`.

Inputs and outputs

- No file, table or database access detected.

Error handling and operations

It sets a return code for the scheduler.

- Sets RETURN-CODE (signals the batch scheduler): 1 `CSUTLDTC.cbl:98`

Modernization notes

- No elevated modernization risk signals detected.

DBUNLDGS Batch program

Convert with SME review

Source
`app/app-authorization-ims-db2-mq/cbl/DBUNLDGS.CBL` · 198 lines

Roadmap stage
Expert-reviewed core

Run by
UNLDGSAM

Data
IMS-DB

Platform
IMS

Parsed
67 statements, 135 data items, 15 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
`ad1e0172-0d27-5602-b080-6e50b0b1750d`

Review
Pending SME review

Our recommendation: Convert with SME review

- Stores years as two digits and applies a century rule that must be made explicit.
- Uses IMS; the data model must be agreed with your data architects.

Purpose

DBUNLDGS is a batch program using IMS. The source has no header describing its purpose; intent below is derived from its structure and data access `DBUNLDGS.CBL:1`.

How it works

It is started by batch job `UNLDGSAM`.

Main processing steps, in source order: main para `DBUNLDGS.CBL:164`; initialize `DBUNLDGS.CBL:182`; find next auth summary `DBUNLDGS.CBL:216`; find next auth dtl `DBUNLDGS.CBL:263`; insert parent seg gsam `DBUNLDGS.CBL:300`; insert child seg gsam `DBUNLDGS.CBL:319`.

Inputs and outputs

- IMS-DB (ims): access `DBUNLDGS.CBL:222`

Error handling and operations

It sets a return code for the scheduler. Writes 15 DISPLAY messages (operators may rely on them).

- Sets RETURN-CODE (signals the batch scheduler): 1 `DBUNLDGS.CBL:362`
- Depends on the current date or time `DBUNLDGS.CBL:185`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

PAUDBLOD Batch program

Convert with SME review

Source
app/app-authorization-ims-db2-mq/cbl/PAUDBLOD.CBL · 251 lines

Roadmap stage
Expert-reviewed core

Run by
LOADPADB

Data
AWS.M2.CARDDEMO.PAUTDB.CHILD.FILEO, AWS.M2.CARDDEMO.PAUTDB.ROOT.FILEO, IMS-DB

Platform
IMS

Parsed
88 statements, 125 data items, 17 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
732b7333-eb1c-542d-bfd2-5f4cae661643

Review
Pending SME review

Our recommendation: Convert with SME review

- Stores years as two digits and applies a century rule that must be made explicit.
- Uses IMS; the data model must be agreed with your data architects.

Purpose

PAUDBLOD is a batch program using IMS. The source has no header describing its purpose; intent below is derived from its structure and data access PAUDBLOD.CBL:1.

How it works

It is started by batch job LOADPADB.

Main processing steps, in source order: main para PAUDBLOD.CBL:169; initialize PAUDBLOD.CBL:190; read root seg file PAUDBLOD.CBL:222; insert root seg PAUDBLOD.CBL:242; read child seg file PAUDBLOD.CBL:269; insert child seg PAUDBLOD.CBL:292; insert ims call PAUDBLOD.CBL:318.

Inputs and outputs

- IMS-DB (ims): access PAUDBLOD.CBL:244
- INFILE1 (file): read PAUDBLOD.CBL:26
- INFILE2 (file): read PAUDBLOD.CBL:32

Error handling and operations

It checks file I/O results and sets a return code for the scheduler. Writes 26 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (2 status fields declared): 8 PAUDBLOD.CBL:202
- Sets RETURN-CODE (signals the batch scheduler): 1 PAUDBLOD.CBL:365
- Depends on the current date or time PAUDBLOD.CBL:193

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

PAUDBUNL

Batch program

Convert with SME review

Source
`app/app-authorization-ims-db2-mq/cbl/PAUDBUNL.CBL` · 207 lines

Roadmap stage
Expert-reviewed core

Run by
UNLDPADB

Data
AWS.M2.CARDDEMO.PAUTDB.CHILD.FILEO, AWS.M2.CARDDEMO.PAUTDB.ROOT.FILEO, IMS-DB

Platform
IMS

Parsed
73 statements, 115 data items, 11 paragraphs

Rehost as is
Ready with platform support
IMS DL/I needs IMS support on the target runtime

Evidence record
`4ac1eb06-b907-5615-84aa-e1b1021a4e14`

Review
Pending SME review

Our recommendation: Convert with SME review

- Stores years as two digits and applies a century rule that must be made explicit.
- Uses IMS; the data model must be agreed with your data architects.

Purpose

PAUDBUNL is a batch program using IMS. The source has no header describing its purpose; intent below is derived from its structure and data access `PAUDBUNL.CBL:1`.

How it works

It is started by batch job `UNLDPADB`.

Main processing steps, in source order: main para `PAUDBUNL.CBL:157`; initialize `PAUDBUNL.CBL:173`; find next auth summary `PAUDBUNL.CBL:207`; find next auth dtl `PAUDBUNL.CBL:253`.

Inputs and outputs

- IMS-DB (ims): access `PAUDBUNL.CBL:213`
- OUTFIL1 (file): write `PAUDBUNL.CBL:26`
- OUTFIL2 (file): write `PAUDBUNL.CBL:32`

Error handling and operations

It checks file I/O results and sets a return code for the scheduler. Writes 15 DISPLAY messages (operators may rely on them).

- FILE STATUS checks (2 status fields declared): 4 `PAUDBUNL.CBL:187`
- Sets RETURN-CODE (signals the batch scheduler): 1 `PAUDBUNL.CBL:313`
- Depends on the current date or time `PAUDBUNL.CBL:176`

Modernization notes

- Packed-decimal / COMPUTE arithmetic — target code must use exact decimal types with matching rounding; verify with old-vs-new output comparison.
- IMS DL/I database calls — hierarchical data needs a mapping strategy before code conversion.

How AI output is controlled

For your risk, compliance and audit teams.

Grounded in your code
Each AI-written profile records the exact files and line ranges it was given, and its statements cite those lines.

Versioned instructions
The prompt is versioned and fingerprinted (program_doc v1.1.0), so any change to how the AI is instructed is visible.

Human sign-off
Nothing is final until a named expert approves it. Every approval is recorded and signed.

Tamper-evident log
Records are chained; altering one after the fact breaks the chain. Status: verified — 61 records verified.

Stays in your environment
The engine runs in your own cluster. Source code goes only to the AI service in your own cloud tenant.

Exactly which code
Every delivered file is fingerprinted (SHA-256). Estate digest `6d22ed24e422fec2…` covers 205 analysed files; the full list is in source-manifest.csv.

Review status
0 of 60 documents and rules approved by a named expert; 60 awaiting review.

Model used
offline / deterministic-template-v1 — a deterministic writer used for this sample; production runs use your Azure OpenAI deployment.

Static analysis of COBOL programs, copybooks, JCL, CICS definitions and DB2 table definitions; no code was run. Calls, batch steps, online transactions and data access were traced into a dependency map. Anything the analysis could not see or confirm is listed under “What this assessment could not see” rather than guessed. Recommendations follow fixed rules applied to the same measurements for every program, so the same code always produces the same plan.