z/OS ADMINISTRATION

A Practical Batch Job Troubleshooting Workbook

Case study • JCL/SDSF investigation • Decision checklist • Knowledge check

Estimated time: 35–50 minutes | Audience: aspiring and working z/OS administrators, operations staff and mainframe support professionals

What you will practise

Investigate a failed or delayed batch job systematically, gather evidence from job output, distinguish symptoms from causes, and decide whether to rerun, escalate or hold the job. The workbook uses a fictional scenario and does not require access to a live mainframe.

1. Scenario: Overnight settlement is delayed

A retail company runs an overnight settlement stream. JOB SETTLE01 extracts transaction records, validates them, updates a settlement dataset, and generates a reconciliation report. At 02:15, the operations dashboard shows the job did not complete as expected. A downstream report is waiting for its output.

Observation	What you know—and what you do not know
Job status shows an error or non-zero completion code.	A failure or warning occurred; the code alone does not identify the root cause.
Downstream job has not started.	It may be waiting for a dependency; do not assume it should be forced to run.
A previous run completed successfully yesterday.	Yesterday's success does not prove today's inputs, environment or dependencies are identical.
Business asks for an immediate rerun.	First check whether any step partially updated data or produced duplicate outputs.

Learning has never been so easy

Your objective

Identify the first failing step, collect evidence, assess data impact, and recommend a safe next action using your site's runbook and escalation rules.

2. A disciplined first-response workflow

Step	Action
1. Confirm the job identity	Record job name, job ID, submit time, owner, status and completion/return code. Make sure you are reviewing the correct run.
2. Inspect spool output	Review the job log and relevant output, including JESMSGLG, JESJCL and JESYSMSG where available. Find the earliest meaningful error, not just the final message.
3. Find the first failing step	Compare step names and return codes. Check whether a step was bypassed, failed during JCL conversion, ended abnormally, or completed with a warning.
4. Check inputs and dependencies	Verify expected input datasets, predecessor jobs, scheduling conditions, file arrival, space availability, permissions and relevant system messages.
5. Assess side effects	Determine whether earlier steps wrote or updated data, created files, sent messages or produced partial output. Check idempotency and recovery instructions.
6. Choose a controlled action	Rerun only when the cause is understood and the runbook confirms it is safe. Otherwise hold dependent work and escalate with evidence.
7. Verify completion	Confirm the correct return code, expected output datasets, record counts/reconciliation, downstream readiness and monitoring status.

Important safety rule

Do not blindly rerun, cancel, release, delete output, or override a job. Follow approved site procedures and obtain the required authorization, especially when a job may have updated production data.

© Copyright protected

3. Read job status and return codes carefully

A return code is a clue, not a complete diagnosis. Interpret it using the program's documented conventions and the site's operational standards. A non-zero code may be an accepted warning in one workflow and a failure in another. An abend, JCL error or security error requires different investigation paths.

Signal	Possible interpretation	Next check
CC 000	Step/job reports normal completion; still verify expected outputs.	Output presence, counts, reconciliation and downstream readiness.
CC 004	May represent a warning or minor condition, depending on program/site rules.	Messages, documented acceptable-code policy and output validity.
CC 008 or higher	Often indicates a more significant condition, but meaning is application-specific.	First failing step, error messages, input quality and application rules.
JCL ERROR	Job failed validation/conversion before normal program execution.	JESJCL/JESMSGLG, syntax, dataset references and procedure overrides.
ABEND	Abnormal termination occurred during execution.	Abend code, failing step, dump/logs and known-error runbook.
SEC ERROR	A security check may have prevented an action.	Resource name, identity, authorization path and security team process.

4. Worked example: what should you do?

Fictional evidence collected: SETTLE01 shows a non-zero completion code. Step VALIDATE completed; step UPDATE ended abnormally; the reconciliation report was not created. The runbook says UPDATE may modify the settlement dataset before a later error is detected. No approved restart point has yet been confirmed.

Unsafe response: immediately resubmit the whole job because the report is missing.

Safer response: hold downstream processing; capture the job ID, failing step, return/abend code and relevant messages; check whether UPDATE changed the dataset; consult the restart/recovery runbook and application owner; rerun only after the recovery point and data integrity are confirmed.

Your decision

What evidence would you need before approving a rerun? List at least four items.

1. _____
2. _____
3. _____
4. _____

5. Troubleshooting checklist

- ■ Correct job name, job ID and run timestamp confirmed.
- ■ Status and return code recorded; first failing step identified.
- ■ Relevant spool output and system/application messages reviewed.
- ■ Input datasets, predecessor jobs, permissions and space checked.
- ■ Partial updates, duplicate processing and side effects assessed.
- ■ Runbook, restart point and required approval confirmed.
- ■ Rerun or escalation decision documented with evidence.
- ■ After recovery, output content and downstream readiness verified.

© Copyright protected

6. Knowledge check

Q1. Why is the first failing step usually more useful than the final job code?

Q2. Does CC 000 prove the business process is correct?

Q3. Why should you check for partial updates before rerunning?

Q4. Which output can help reveal JCL and system messages?

Q5. When should you escalate rather than rerun?

Suggested answers

1. It narrows investigation to the earliest point where expected behaviour diverged; later messages can be consequences.

2. No. Validate expected output, counts, data integrity and downstream readiness.

3. A rerun may duplicate transactions, overwrite valid data or create inconsistent results.

4. Common spool outputs include JESJCL, JESMSGLG and JESYSMSG, depending on the job and environment.

5. When cause, data impact, restart safety, authorization or recovery procedure is unclear, or site policy requires escalation.

7. Incident handover template

Job / ID / time: _____
Status / return code / abend: _____
First failing step and key message: _____
Inputs / dependencies checked: _____
Potential data side effects: _____
Actions already taken: _____
Help / approval needed: _____
Next owner and follow-up time: _____

Further learning and source

IBM z/OS Basic Skills — “Doing work on z/OS: How you submit, control and monitor jobs using JCL and SDSF.” This official resource explains JCL’s role and how SDSF is used to monitor and inspect job output: https://www.ibm.com/docs/en/zos-basic-skills?topic=zc-doing-work-zos-how-you-submit-control-monitor-jobs-using-jcl-sdsf

IBM documentation notes that job output can include job status, return codes and spool files such as JESMSGLG, JESJCL and JESYSMSG. Always follow your organisation’s runbook, access controls and change procedures. This workbook is educational and does not replace site-specific operating instructions.

Practical takeaway: investigate, gather evidence, assess data impact, follow the runbook, then act—and verify the outcome.

© Copyright protected