

The Governed SAP Job Recovery Playbook

A reference operating model and nine deployable runbooks for recovering background-job failures across SAP and connected enterprise systems — designed to make recovery repeatable, accountable, and auditable.

Prepared for SAP Basis, IT Operations, and S/4HANA migration teams.

VERSION	STATUS	CLASSIFICATION	OWNER
1.0	Released	Public	BCS Operations

Business Core Solutions · Symphony — the agentic orchestration layer for enterprise operations
India · Australia · Netherlands · United States · United Kingdom · Germany · UAE

00 Executive summary

The Governed SAP Job Recovery Playbook

Symphony • Business Core Solutions

Modern schedulers detect background-job failures reliably. The operational risk has moved downstream — to what happens *after* detection. The review, the restart-or-skip decision, the authorisation, the ticket, and the evidence capture remain largely manual, and they are where mean time to recover (MTTR) erodes and audit trails become inconsistent.

That manual recovery layer carries three recurring costs. First, recovery does not begin until a qualified person is available, which extends outage duration during off-hours. Second, decisions are made without a consistent control around who may authorise an action and at what threshold. Third, the evidence required to demonstrate the recovery — to an auditor, a regulator, or a post-incident review — is reconstructed after the fact rather than captured as the work is performed.

The exposure increases materially during S/4HANA migration. Inefficient job scheduling is a recognised challenge in the transition, and cutover sequences comprise hundreds of interdependent tasks executed inside fixed downtime windows. Period-close calendars are equally unforgiving. In these conditions, an undefined recovery path is not an inconvenience; it is a schedule and compliance risk.

What this document provides

This playbook defines a five-stage governed recovery model and provides nine runbooks covering the failure patterns operations teams encounter most. Each runbook is written to be adopted directly into an existing operations or ITSM practice: complete the bracketed parameters, assign the roles, and publish it to the on-call team. The closing section describes how Symphony operationalises the same model continuously and at scale.

Reading guidance. Transaction codes (e.g. `SM37` , `ST22` , `BD87`) and report or job names (e.g. `RFF1105`) are illustrative and shown in monospace; substitute the standards in use across your landscape. Bracketed fields such as `[threshold]` denote parameters to be set per environment.

01 The governed recovery model

The Governed SAP Job Recovery Playbook

Symphony • Business Core Solutions

Every runbook in this playbook follows the same five-stage model. Applied consistently, it converts recovery from an individual effort into a controlled, repeatable process with a complete record of who acted, what they did, and why.

1 • Monitor

Observe the job and its dependencies — not only the alert — and establish the business impact.

2 • Recommend

Determine the appropriate action — restart, skip, escalate — with the supporting context.

3 • Approve

Route the decision to the accountable role at the defined authorisation threshold.

4 • Recover

Execute the approved action — and only the approved action.

5 • Audit

Record the failure, decision, approval, action, and outcome as the work is performed.

Operating principles

- **Posting and financial jobs are never retried without review.** A partial posting must be reconciled before any re-run is authorised.
- **Every action has a named, recorded approver** — including a decision to auto-retry, which is logged as a deliberate policy outcome.
- **Evidence is captured during recovery**, not reconstructed afterwards.
- **Absence is a signal.** A job that should have run and did not constitutes an incident.

How to adopt these runbooks

1. Import the relevant runbook into your operations wiki or ITSM knowledge base.
2. Complete the `[bracketed]` parameters — thresholds, owners, approvers, systems, and windows.
3. Confirm the roles against the governance model in Appendix A and brief the on-call team.
4. Treat any parameter you cannot complete as a finding: it marks a control that does not yet exist in your environment.

02 Contents

The Governed SAP Job Recovery Playbook

Symphony • Business Core Solutions

Runbooks

- 01 Overnight / off-hours job failure
- 02 Long-running job delaying downstream processes
- 03 IDoc / interface failure
- 04 Recurring job silently removed from the schedule
- 05 Month-end / period-close job failure
- 06 Maintenance-window suspension and restart
- 07 Cutover batch-dependency cascade (S/4HANA migration)
- 08 Post-go-live reconciliation / failed data load
- 09 Repeat failure with disconnected ticket, owner, and logs

Appendices

- A Governance model and RACI
- B Severity, SLA, and escalation framework
- C Transaction-code reference
- D Operating the model at scale

03 At a glance

The nine runbooks summarised. Use this matrix to identify the relevant runbook, the decision it turns on, and the role accountable for authorising recovery.

RUNBOOK	DETECTION TRIGGER	DECISION IT TURNS ON	AUTHORISER	COST IF UNGOVERNED
01 · Overnight / off-hours failure	Job Cancelled outside staffed hours	Transient → controlled retry; data error → hold and escalate	On-call owner; threshold approver	No owner until the next shift; MTTR extends
02 · Long-running job, downstream delay	Runtime breach vs. baseline; SLA at risk	Extend / terminate and restart / reroute	Business owner	No one authorised to act; SLA slips
03 · IDoc / interface failure	Error status in WE02 / WE05	Reprocess / correct-at-source / escalate	Process owner	Alert detached from business impact
04 · Job silently removed	Time-since-last-run breach (absence)	Reinstate schedule and scope backfill	Job owner	Absence raises no flag; loss is invisible
05 · Period-close failure	Close-critical job Cancelled in window	Re-run vs. manual correction — never a blind retry	Dual: IT + Finance	Recovery not provable to an auditor
06 · Maintenance suspension / restart	Planned window; jobs suspended	Restart sequence and do-not-resume set	Window owner	A job silently fails to resume
07 · Cutover dependency cascade	Cutover task fails; chain at risk	Halt-and-fix / re-sequence / rollback vs. window	Cutover lead	Cascade recognised too late
08 · Post-go-live reconciliation	Variance beyond tolerance / load fails	Reprocess / correct-at-source / quarantine	Data owner	Unverified data enters reporting
09 · Repeat / disconnected failure	≥ 3 correlated occurrences	Manage as a problem; fix the cause	Application / CI owner	The loop never closes

Each runbook expands this row into a full procedure — detection, triage, decision criteria, approval, execution, validation, and evidence.

RUNBOOK 01 **Overnight / off-hours job failure**

The Governed SM Job Recovery Playbook

Symphony → Business Core Solutions

Objective. Recover a business-critical batch failure that occurs outside staffed hours within its SLA, under a defined owner, with a complete evidence record.

Accountability question: Who is accountable for recovery before the next shift begins?

SEVERITY	TYPICAL OWNER	SLA / WINDOW	DETECTION
SEV-1	On-call Basis	AP cutoff · 06:00	SM37 · CCMS

A critical batch job — for example the payment proposal RFF110S — cancels between [22:00] and [06:00] with no operator on shift. Without a defined owner and authorisation path, recovery does not begin until staff return, extending the outage and leaving the decision undocumented.

PREREQUISITES On-call access to SM37, ST22, and SM21; the dependency map for the failed job; and the documented restart-authorisation threshold for financially material jobs.

DETECTION The job enters status **Cancelled** in SM37; CCMS (RZ20), BPMon, or Focused Run raises an alert. Record the return code, the failing step, and the start and end times.

TRIAGE

- Retrieve the job log (SM37 → Job Log) and read the terminating message.
- If a runtime dump is referenced, classify it in ST22 — e.g. TIME_OUT, DBIF_RSQI_SQL_ERROR, TSV_TNEW_PAGE_ALLOC_FAILED.
- Review SM21 for database, lock, or memory events at the failure time.
- Identify downstream dependents and the binding SLA.

DECISION	Transient error (lock, resource, deadlock, timeout)	Authorise a controlled retry, up to [n] attempts. On success, record and close; if exhausted, escalate.
	Data or configuration error	Suppress retry. Hold the job, freeze dependents, and escalate to the on-call owner; a blind restart will re-fail or post incorrect data.

APPROVAL Notify the on-call owner [role]. A restart of a financially material job above [threshold] requires documented approval from [approver], recorded with identity and timestamp.

RECOVERY

- Remediate the root cause (clear the duplicate, correct the variant, release resources).
- Restart from the last restart-capable step. **For a payment run (F110), resume via Edit → Payments → After Termination → Draw up again** — never an SM37 "repeat", which re-pays already-dispatched items (double-payment risk). Do not create a parallel job that bypasses the original's controls.
- Release held dependents in sequence.

VALIDATION Job completes with return code 0; dependents released and progressing; the business artefact (e.g. payment proposal) is produced.

EVIDENCE Record the failure, dump reference, decision rationale, approver, action taken, and outcome to the recovery record.

COMMON CAUSES Duplicate key on re-run (DBIF_RSQI_SQL_ERROR); memory exhaustion on high volumes (TSV_TNEW_PAGE_ALLOC_FAILED); lock contention with a parallel job; an expired or amended variant.

CONTROL GAP

Without a defined owner and authorisation threshold, the failure persists until staff return — extending MTTR and leaving the decision undocumented.

WITH SYMPHONY

Symphony classifies the error, suppresses unsafe retries, obtains and records approval, executes the restart, releases dependents, and writes the evidence — without manual effort outside the approval step.

Parameters: [off-hours window] · [retry count] · [on-call role] · [restart approver] · [threshold]

RUNBOOK 02 Long-running job delaying downstream processes

Objective. Protect a dependent SLA when a job exceeds its expected runtime, through a timely, authorised decision to extend, terminate, or reroute.

Accountability question: Who is authorised to wait, terminate, or reroute — and on whose authority?

SM37

SM50 / SM66

SM12

Runtime breach

DETECTION Runtime exceeds the historical baseline by `[+30%]` (warning) or `[+100%]` (critical), per `SM37` duration against trend. A dependent process is approaching its SLA deadline.

- TRIAGE**
- Determine whether the job is progressing or stalled: inspect the active work process in `SM50 / SM66` — CPU, sequential read, or an enqueue/database wait.
 - Check for lock contention (`SM12`) and long-running statements (`ST04` / DBACOCKPIT).
 - Identify the blocked dependents and the binding SLA.

DECISION	Progressing; will complete within SLA	Extend the window with notice to the dependent owner; continue monitoring.
	Stalled; no measurable progress	Terminate and restart from the last <code>[checkpoint]</code> ; do not restart from the beginning where a checkpoint exists.
	SLA breach unavoidable	Reroute or re-sequence dependents and re-baseline the ETA with the owner.

APPROVAL Any SLA-impacting action (terminate, extend beyond deadline, reroute) is acknowledged by the business owner `[role]` before execution.

- RECOVERY**
- Where terminating, stop the active process (`SM50 → Cancel without core`) or cancel in `SM37` .
 - Restart from the checkpoint variant and rebalance the queue so dependents resume in order.

VALIDATION Job completes or is cleanly rerouted; the dependent SLA is met or formally re-baselined; the queue is stable.

EVIDENCE Record the runtime breach and baseline, the decision, the approver, and the SLA outcome.

CONTROL GAP

With no one authorised to act, the team defaults to waiting and the SLA slips without an accountable decision.

WITH SYMPHONY

Symphony detects the breach early, distinguishes progress from stall, scores the options against the SLA, secures the owner's acknowledgement, and rebalances the queue.

Parameters: `[baseline]` · `[warning/critical %]` · `[SLA deadline]` · `[checkpoint]` · `[business owner]`

RUNBOOK 03 IDoc / interface failure

Objective. Resolve an interface failure with reference to its business consequence, ensuring the correct owner authorises any business-impacting reprocessing.

Accountability question: Who connects the technical error to its business consequence?

WE02 / WE05

BD87

SM58

Status 02 / 51

DETECTION An IDoc enters an error status in WE05 (outbound) or WE02 (inbound). Common codes: 02 (port/ALE), 29 (ALE service), 51 (application document not posted), 68 (locked). Transactional RFC failures appear in SM58.

- TRIAGE**
1. Read the status record text for the precise field or segment error.
 2. **Associate the IDoc with its business object** — the delivery, invoice, or order, the counterparty, the value, and the deadline. This association is the step most often omitted.
 3. Classify the cause: connectivity, master data, or application/data.

DECISION	Connectivity (port / RFC)	Restore the connection (SM59 , WE21), then reprocess unchanged.
	Master data or application error	Correct at source, then reprocess. Do not force-post a malformed IDoc.
	Business decision required	Escalate to the process owner with the business impact attached.

APPROVAL Business-impacting reprocessing is authorised by the process owner [SD/MM/FI owner] accountable for the consequence.

- RECOVERY**
1. Inbound: reprocess via BD87 , filtered by status, message type, and partner. For status 68, unlock and reset before posting.
 2. Outbound: re-trigger via WE14 or the application and confirm transmission.

VALIDATION The IDoc reaches a success status (53 inbound; 12/16 outbound) and the business document is confirmed.

EVIDENCE Record the error code and text, the business impact, the action, the approver, and the resolution.

CONTROL GAP

The alert is handled as a technical status without its business context, and a customer or supplier impact is discovered late.

WITH SYMPHONY

Symphony links the IDoc to the affected document and counterparty, attaches the business impact, routes the correct owner, reprocesses, and records a single trail.

Parameters: [message types in scope] · [partner profiles] · [process owner by area]

RUNBOOK 04 Recurring job silently removed from the schedule

Objective. Detect and remediate a periodic job that has stopped running without raising a failure, and backfill the missed work under controlled scope.

Accountability question: Who notices the job that is no longer running?

SM37

SM36 / SM37C

Status: Absent

DETECTION	This failure produces no event; the signal is absence . Monitor time since last successful run against the expected cadence, and alert when <code>now - last_success > interval × [1.5]</code> . Maintain an authoritative register of expected jobs and frequencies.
TRIAGE	1. Confirm the job is genuinely unscheduled (<code>SM37</code> over a wide date range). 2. Establish when and why it stopped (change/transport history, <code>SM37C</code> standard-job configuration). 3. Quantify the gap — runs missed and output not produced.
DECISION	Reinstate the schedule and define the backfill scope: which missed runs must be executed, in what order, and whether backfilling is safe — some jobs must run once only, or for a specific period.
APPROVAL	The owner <code>[role]</code> confirms the backfill scope before any catch-up execution.
RECOVERY	1. Re-create and release the periodic job (<code>SM36</code>) with the correct variant, calendar, and start condition. 2. Execute the approved backfill in date order, validating each run before the next.
VALIDATION	The job is released with its next run scheduled; the backfill is complete and any side effects (e.g. table growth) are normalised.
EVIDENCE	Record when and why it stopped, who reinstated it, the backfill performed, and the scope approver.

CONTROL GAP

A job that never runs cannot raise a failure; the loss remains invisible until its downstream effect becomes material.

WITH SYMPHONY

Symphony maintains an expected-job baseline and alerts on absence, then reinstates the schedule and backfills against an approved scope.

Parameters: `[expected-job register]` · `[absence multiplier]` · `[backfill approver]`

RUNBOOK 05 **Month-end / period-close job failure**

The Governed SAP Job Recovery Playbook

Symphony > Business Core Solutions

Objective. Recover a close-critical job within the close window without compromising financial integrity, under dual authorisation and with audit-ready evidence.

Accountability question: Who authorises recovery against a live close calendar — and demonstrates it to the auditor?

SM37 SAPP010 / F.16 FAGLB03 / FAGLL03 **Close-critical**

DETECTION	A close-critical job (e.g. GL reconciliation / balance audit trail <code>SAPP010</code> , balance carryforward <code>F.16</code> / <code>FAGLGVTR</code> , a ledger posting run) cancels inside the close window. Classify immediately as close-critical / SEV-1.					
TRIAGE	1. Freeze dependent postings to prevent partial data entering the ledger during assessment. 2. Determine whether the job posted partially — confirm committed document numbers. This materially changes the recovery path. 3. Retrieve the dump (<code>ST22</code>) and the failing step.					
DECISION	<table><tr><td>Failed before any posting</td><td>Re-run after remediating the cause (e.g. memory parameters, lock).</td></tr><tr><td>Partial posting occurred</td><td>Do not retry. Reconcile with Finance — identify posted versus not-posted and decide manual correction or a scoped re-run for the remainder.</td></tr></table>		Failed before any posting	Re-run after remediating the cause (e.g. memory parameters, lock).	Partial posting occurred	Do not retry. Reconcile with Finance — identify posted versus not-posted and decide manual correction or a scoped re-run for the remainder.
Failed before any posting	Re-run after remediating the cause (e.g. memory parameters, lock).					
Partial posting occurred	Do not retry. Reconcile with Finance — identify posted versus not-posted and decide manual correction or a scoped re-run for the remainder.					
APPROVAL	Dual authorisation: <code>[IT owner]</code> (technical safety) and <code>[Finance owner]</code> (financial correctness). Both approvals are recorded with timestamps.					
RECOVERY	1. Execute only the approved re-run or correction. 2. Reconcile (<code>FAGLB03</code> balances, <code>FAGLL03</code> line items under New GL — <code>FBL3N</code> only for classic GL) and confirm the period ties out.					
VALIDATION	Period integrity is confirmed with no duplicate or partial postings; the close completes within the window.					
EVIDENCE	Assemble an evidence pack: control point, both approvals, action, and reconciliation result.					

CONTROL GAP

The recovery cannot be demonstrated to an auditor, and an unreviewed retry risks a duplicate posting.

WITH SYMPHONY

Symphony flags the job as close-critical, suppresses retry on posting jobs, enforces dual authorisation, and assembles the evidence pack automatically.

Parameters: `[close deadline]` · `[close-critical job register]` · `[IT owner]` · `[Finance owner]`

RUNBOOK 06 Maintenance-window suspension and restart

Objective. Suspend and reinstate background processing across a planned window with full assurance that every job resumes, in the correct order.

Accountability question: Who confirms that every suspended job has resumed — and in sequence?

BTCTRS1 / 2

SM37

SM65

Planned downtime

PREPARATION	1. Snapshot the full set of suspended jobs and their dependency order from <code>SM37</code> . 2. Suspend batch processing cleanly with <code>BTCTRS1</code> ahead of downtime. 3. Identify jobs that must not auto-resume — one-time runs, or jobs whose window has elapsed.
DECISION	Define the restart sequence and prerequisites (which systems and interfaces must be healthy first), and the jobs to skip, resume, or reschedule.
APPROVAL	The window owner <code>[role]</code> confirms system health — database, interfaces, open transports — before batch processing is released.
RECOVERY	1. Reactivate jobs with <code>BTCTRS2</code> once health is confirmed. 2. Resume in the approved sequence, validating each job and its dependents before the next tier. 3. Confirm batch-system health in <code>SM65</code> .
VALIDATION	Reconcile the suspended set against the resumed set — every job accounted for, none left suspended or released-but-not-run.
EVIDENCE	Record the suspended set, the restart sequence, the health sign-off, and any job that did not return.

CONTROL GAP

A suspended job fails to resume and is not noticed until its missing output surfaces, often days later.

WITH SYMPHONY

Symphony snapshots the suspended set, gates restart on a health sign-off, resumes in order, and reconciles suspended against resumed to zero.

Parameters: `[window owner]` · `[restart sequence]` · `[do-not-resume list]` · `[health checks]`

RUNBOOK 07 Cutover batch-dependency cascade (S/4HANA migration)

Objective. Contain a failure within an interdependent cutover sequence and select a recovery path that protects the downtime window, under change-freeze governance.

Accountability question: Who recognises the cascade forming and halts the chain in time?

Migration Cockpit

STMS

Cutover plan

Window at risk

DETECTION A task in the cutover sequence reports failure — a data load, a migration object, or a transport import (**STMS**). The two governing figures are the critical path and the time remaining in the window; track both continuously.

TRIAGE

1. Map the downstream chain immediately — what is blocked now, what is at risk next, what remains safe.
2. Estimate time-to-remediate against the time remaining in the window.

DECISION	Fast remediation; window has slack	Halt and fix: pause the chain, correct, resume from the failure point.
	Independent tasks remain	Re-sequence: progress unaffected branches while remediating the failed one.
	Remediation exceeds remaining window	Roll back to the last [checkpoint] and invoke the fallback plan.

APPROVAL The cutover lead **[role]** authorises the selected path within the change-freeze rules. No remediation proceeds outside the freeze without this authorisation.

RECOVERY

1. Execute the approved path.
2. Re-validate the sequence from the failure point forward; do not assume downstream tasks remain valid.

VALIDATION Sequence integrity is restored and the window status is confirmed: on track, re-baselined, or fallback invoked.

EVIDENCE Record the cascade, the decision, the time impact, the window status, and the authoriser to the cutover record.

CONTROL GAP

The cascade is recognised too late; by the time it is evident, the downtime window is already lost.

WITH SYMPHONY

Symphony maps the dependency chain in real time, scores the options against the remaining window, and routes the cutover lead within the freeze rules.

Parameters: **[downtime window]** · **[critical path]** · **[rollback checkpoint]** · **[cutover lead]**

RUNBOOK 08 Post-go-live reconciliation / failed data load

Objective. Resolve a post-migration load failure or reconciliation variance while protecting reporting integrity until the data is proven correct.

Accountability question: Who owns the correction and demonstrates data integrity after go-live?

FAGLB03

MB5B

WE02

Reconciliation variance

DETECTION A reconciliation variance exceeds tolerance `[materiality]` , or a load job fails, under the S/4HANA data model. Compare against the cutover reconciliation baseline — record counts and control totals.

TRIAGE

1. Isolate scope: which records, which interface or object, and which downstream reports are affected.
2. Compare control totals, legacy against S/4 (`FAGLB03` for GL, `MB5B` for stock, open-item lists for AP/AR).
3. Quarantine the affected figures so reporting does not proceed on unverified data.

DECISION	Load or mapping error	Reprocess the corrected load for the affected scope.
	Source data incorrect	Correct at source and reload; do not patch the target in place.
	Not resolvable immediately	Maintain the quarantine and protect reporting until resolved.

APPROVAL The data owner `[role]` authorises the correction method and owns the resulting integrity statement.

RECOVERY

1. Apply the approved correction to the isolated scope only.
2. Re-run reconciliation and confirm a clean tie-out within tolerance before releasing reporting.

VALIDATION Reconciliation ties out within tolerance and the affected reports are released from quarantine.

EVIDENCE Record the root cause, the correction, the approver, and the passing reconciliation as evidence.

CONTROL GAP

Unverified data enters reporting before integrity is demonstrated, eroding confidence in the new system at go-live.

WITH SYMPHONY

Symphony isolates the scope, quarantines reporting, routes the data owner, and re-runs reconciliation to a recorded clean tie-out.

Parameters: `[tolerance / materiality]` · `[reconciliation baselines]` · `[data owner]`

RUNBOOK 09 Repeat failure with disconnected ticket, owner, and logs

The Governed SaaS Job Recovery Playbook

Symphony > Business Core Solutions

Objective. Convert a recurring incident into a managed problem by correlating its history, addressing the root cause, and verifying the fix.

Accountability question: Who closes the loop when the same job fails repeatedly and nothing is connected?

SM37

ITSM

Problem management

Recurring

DETECTION	Failures are correlated by job name and error signature over a rolling window [14 days] ; a threshold of [≥3 occurrences] promotes the incident to a problem.
TRIAGE	1. Consolidate the failure history — all SM37 occurrences, job logs, and dumps — into a single timeline. 2. Join the timeline to the related ITSM tickets and the owning team or configuration item. 3. Identify the recurring root cause, distinct from the latest symptom.
DECISION	Manage as a problem rather than a series of incidents — address the cause, not the run. Keep the relevant runbook in force as an interim control until the permanent change is deployed.
APPROVAL	The application or configuration-item owner [role] approves the permanent change through normal change control.
RECOVERY	1. Deploy the permanent change via change management. 2. Confirm the failure pattern ceases over [n] subsequent runs before closing the problem.
VALIDATION	No recurrence over the agreed observation period; the problem record is closed with verification evidence.
EVIDENCE	Record the consolidated history, root cause, change, approver, and verification in a single trail.

CONTROL GAP

With ticket, log, and ownership held separately, each occurrence is treated as new and the underlying cause is never resolved.

WITH SYMPHONY

Symphony correlates ticket, log, and owner into one record, surfaces the recurring cause, and maintains an interim control until the fix is verified.

Parameters: [correlation window] · [occurrence threshold] · [CI / application owner] · [observation runs]

A Governance model and RACI

Governed recovery depends on roles agreed in advance of any failure. Complete this matrix for your organisation and publish it alongside the runbooks. Adjust the authorisation thresholds to your delegation-of-authority policy.

RECOVERY ACTIVITY	BASIS / OPS ON-CALL	APPLICATION / PROCESS OWNER	BUSINESS / FINANCE OWNER	CHANGE / CUTOVER LEAD
Detect and triage	R	C	I	I
Classify and recommend action	R	C	I	I
Approve standard restart (below threshold)	A	C	I	—
Approve restart above threshold	C	A	C	—
Approve SLA-impacting action	C	C	A	—
Approve close re-post (dual)	A	C	A	—
Approve cutover path (within freeze)	C	C	I	A
Execute recovery	R	C	I	C
Validate and close evidence record	R	A	I	I

R Responsible A Accountable (single owner) C Consulted I Informed.

B Severity, SLA, and escalation framework

The Governed SAP Job Recovery Playbook

Symphony • Business Core Solutions

A baseline framework. Replace the response and resolution targets with your own service commitments.

SEVERITY	DEFINITION	RESPONSE	RESOLUTION	AUTHORISER
SEV-1 — Critical	Production or financial reporting blocked; close or cutover window at risk.	[5 min]	[1 hr]	Application + Business/Finance
SEV-2 — High	Business-critical job failed; interface down; a single process impacted.	[15 min]	[4 hr]	On-call + Application
SEV-3 — Medium	Non-critical job failed; workaround available; limited impact.	[1 hr]	[24 hr]	On-call

Worked example — month-end close

Document cutoff	[30th, 23:59]
Batch deadline	[1st, 06:00]
Reconciliation SAPF010	complete by [04:00] · SLA 4 hr
Settlement RGSC0000	complete by [05:30] · SLA 1.5 hr
On a miss	Raise SEV-1 and invoke the dual-authorisation path (Runbook 05)

Recommended measures

- **Mean time to detect (MTTD)** — including absence-based detection.
- **Mean time to recover (MTTR)** — by severity and by runbook.
- **Recovery completeness** — the proportion of recoveries closed with a complete evidence record.
- **Recurrence rate** — incidents promoted to problems and resolved at the cause.

c Transaction-code reference

SM36	Define and schedule a background job.
SM37	Job overview and monitor. Statuses: Scheduled, Released, Ready, Active, Finished, Cancelled.
SM37B / SM37C	Extended job selection / standard-jobs configuration.
SM50 / SM66	Work-process overview, local and global.
SM12	Lock entries — enqueue conflicts.
SM21	System log.
ST22	ABAP runtime-error (dump) analysis.
SM65 / RZ20	Background-processing analysis / CCMS alert monitor.
BTCTRNS1 / 2	Suspend / reactivate batch processing for a maintenance window.
WE02 / WE05	IDoc list display, inbound / outbound.
WE20 / WE21	Partner profiles / ports.
BD87	Reprocess inbound IDocs by status, message type, or partner.
WE14	Re-trigger outbound IDoc dispatch.
SM58 / SM59	tRFC monitor / RFC destination maintenance.
F110 • SAPF110S	Automatic payment run (SAPF110S runs it; RFF110S schedules it).
SAPF010 • F.16	GL reconciliation / balance audit trail • balance carryforward (illustrative close jobs).
FAGLB03 / FAGLL03	GL balance display / line-item display (New GL; FBL3N for classic GL).
MB5B	Stock on posting date (inventory reconciliation).
STMS	Transport Management System.
Migration Cockpit	S/4HANA data migration ("Migrate Your Data" / LTMOM).

Job and report names are illustrative; substitute the standards in use. IDoc success statuses: 53 (inbound posted); 12 / 16 (outbound dispatched / confirmed).

D Operating the model at scale

These runbooks define the recovery model. Applied manually, the model depends on the right person executing the right runbook at the right time — frequently outside business hours. Symphony operates the same model continuously and across the estate.

Symphony monitors each job and its dependencies, recommends the appropriate action with full context, routes the decision to the accountable role at the defined threshold, executes the approved action, and records the complete audit trail as the work is performed — from a single orchestration layer spanning SAP and connected enterprise systems. The runbook is the documented intent; Symphony is the means of executing it consistently, every time.

Reported outcomes

Up to 50%	reduction in mean time to recover.
Up to 70%	reduction in audit-preparation effort, as evidence is captured during recovery.
Up to 80%	faster onboarding of jobs into a governed model.

Indicative figures from Symphony Background Job Management deployments; outcomes depend on estate, scope, and operating maturity.

Next step. Arrange a 30-minute governed-recovery walkthrough to review these runbooks against the failure patterns specific to your landscape.
runsymphony.com/governed-recovery · hello@businesscoresolutions.com