

Enterprise Scheduler — Step-by-step Walkthrough

Publish one job for a Vista report and one for Pulse output, run them under an explicit execution owner, add a calendar-aware schedule, and recover bounded missing/failed work with retained lineage.

Use the current tenant, profile and permitted actions. Screenshots illustrate the application; example records and date fields may differ in your environment.

Contents

1. Demonstration story, prepared data and participants
2. Demonstration sequence and visible outcomes
3. Demonstration · Define and preview an operation with an accountable owner
4. Demonstration · Run once and reconcile the producer result
5. Demonstration · Create a calendar-aware one-time or recurring schedule
6. Demonstration · Choose retry, rerun, cancel, pause or retire correctly
7. Demonstration · Govern queue capacity and inspect worker leases
8. Demonstration · Preview and admit only missing historical occurrences
9. Demonstration · Run dependencies and resume from successful checkpoints
10. Demonstration · Verify the complete automation trail
11. Additional demonstration · Access, prerequisites and example values
12. Additional demonstration · Define, publish and run a job
13. Additional demonstration · Step 1 · Published artifact and Scheduler operation bindings
14. Additional demonstration · Step 2 · Create and govern a job definition
15. Additional demonstration · Step 3 · Configure report, dashboard, email and letter operations
16. Additional demonstration · Step 4 · Bind current dataset values to job inputs
17. Additional demonstration · Step 5 · Execute a job and inspect its result
18. Additional demonstration · Step 6 · Inspect and verify execution audit evidence
19. Additional demonstration · Create and operate a schedule
20. Additional demonstration · Step 1 · Business calendars, time zones and DST
21. Additional demonstration · Step 2 · Create a one-time or recurring schedule
22. Additional demonstration · Step 3 · Timeouts, retries, overlaps and misfires
23. Additional demonstration · Step 4 · Queues and workers: govern capacity and execution
24. Additional demonstration · Step 5 · Rerun, cancel, pause, resume and retire

- 25. Additional demonstration · Recover missed work and run dependent batches
- 26. Additional demonstration · Step 1 · Backfill missing schedule occurrences
- 27. Additional demonstration · Step 2 · Batch dependencies across source refresh, Vista and Pulse
- 28. Additional demonstration · Step 3 · Pause and resume a batch using checkpoints
- 29. Additional demonstration · Step 4 · Update, deactivate and retain historical records
- 30. Cross-module handoffs and the record references to retain
- 31. Problem → responsible action → safe continuation
- 32. Close the demonstration and hand over the operating guide

Demonstration story, prepared data and participants

Publish one job for a Vista report and one for Pulse output, run them under an explicit execution owner, add a calendar-aware schedule, and recover bounded missing/failed work with retained lineage.

Worked example and preparation

Prepare eligible published report, dashboard, email and letter versions and a source-refresh connection where needed. Use a small training range and an isolated known failure for recovery; do not stop a shared worker to create a demonstration incident.

Worked record values and expected totals

record_id	status	amount
CASE1001	OPEN	1200
CASE1002	CLOSED	500
CASE1003	OPEN	300

Expected case count is 3 and total amount is 2000. The OPEN group has 2 cases and amount 1500; CLOSED has 1 case and amount 500. Use CASE1001 for a primary-record email/letter. A related-items dataset with two rows for that case should repeat two detail rows, without treating the repeated parent amount as two separate cases. The values must be prepared through the approved source owner before the live exercise.

Responsibilities

Role	Responsibility
Job/schedule maker and reviewer	Previews and governs exact operation, owner, content and timing.
Execution owner	Provides current authority to read inputs and invoke the producer.
Queue/worker operator	Controls bounded admission, leases and service health.
Auditor	Follows run, attempt, producer, checkpoint and recovery references.

Terms used in this module

Term	Meaning
Job definition	What registered operation runs, under whose authority and with what inputs.
Schedule	When eligible job occurrences are admitted.

Term	Meaning
Run / attempt	Execution identity and individual tries within it.
Rerun	A linked new execution, distinct from retrying one attempt lifecycle.
Checkpoint	Recorded successful batch-step progress; the producer owns its internal work.

The procedures below explain operation. Application captures use illustrative records. Record actual demonstration and UAT outcomes separately; a written expected result is not a recorded pass.

CHAPTER 2

Demonstration sequence and visible outcomes

Business journey	Open the detailed steps
Governed job and first producer result	<u>Define and preview an operation with an accountable owner</u> <u>Run once and reconcile the producer result</u>
Calendar timing and bounded operational recovery	<u>Create a calendar-aware one-time or recurring schedule</u> <u>Choose retry, rerun, cancel, pause or retire correctly</u> <u>Govern queue capacity and inspect worker leases</u>
Backfill, dependent batch and audit	<u>Preview and admit only missing historical occurrences</u> <u>Run dependencies and resume from successful checkpoints</u> <u>Verify the complete automation trail</u>

CHAPTER 3

Demonstration · Define and preview an operation with an accountable owner

Open	Who acts
Scheduler → Job Definitions → New job	Job maker and reviewer

Select a registered operation and schema rather than entering unrestricted executable code or arbitrary external URLs.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Business operation	Run published report/dashboard, generate email/letter, refresh connection or another eligible registered operation	External business components must be registered, authorized and schema-bound before selection.

Input shown on screen	Enter or select	Explain its significance
Published content	The eligible artifact for the operation	The selector pins version/revision/hash rather than only a title.
Execution identity / domain	The current authorized service/user context offered by the platform	The owner needs authority over every input and producer.
Inputs / bindings	Typed constants and eligible dataset values	A dataset binding resolves current data; do not place secrets in a constant.
Timeout / attempts / overlap	Policy suited to producer side effects	The inspected bounds are 1–86400 seconds, 1–10 eligible attempts and bounded concurrency 1–32.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Choose the operation before its content.	The correct channel/type-specific selector and inputs appear.
2	Select published content, execution identity/domain and required inputs.	The job's authority and dependencies are explicit.
3	Save and Preview the job.	Input resolution, producer binding and access are checked without executing business work.
4	Inspect empty/missing input and owner-access failures before submission.	The first unresolved boundary is visible.
5	Have the independent reviewer preview and decide the saved revision.	The reviewed job matches current input resolution.
6	Publish and inspect the exact version.	Run/schedule actions can use the controlled definition.

SCREEN 2

Job preview resolves inputs without executing the business operation

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

🔔

💬

🔄

Quick Links

SM

Studio Maker

Platform Demo Maker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 3

Job Definitions

Schedules

Job Runs

Pulse

Vista

Nexora

Active Access Profile

Platform Demo Maker

Menus reflect this profile's permissions.

MediFlex > Core > Job Definitions

View A B C D E Overview

Back

Core

Preview · Browser scheduled artifact qbox 1792317514110

Current input values and the registered operation are shown. Preview does not execute the business operation.

Saved version

Preview valid until

Timeout

1 · Revision 1

10/18/2026, 3:43:45 PM

300 seconds

Search loaded records...

Input	Current Value
reason	Scheduled reporting browser acceptance
reportId	becfa741-93cf-4fbe-a2a9-455f6321af70
versionId	22507ced-4e69-46c2-a9be-c28793a9bef6
contentHash	80e46d065ba9d05c6e14752dcdbcb692c75540a2a26ae0092c8ebbdd5a5496f9
expectedRevision	4

Showing 1-5 of 5

Return to the job row to submit or review. Changed data, permissions or draft details require a new preview.

- Preview establishes current saved-definition eligibility.
- It does not generate the report; an actual run comes later.

Evidence to show before continuing

Show registered operation, owner, pinned content and non-executing preview.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
An external operation is absent	Have its owner register and govern the operation contract; Scheduler is not an unrestricted remote-code launcher.
Preview succeeds but execution later denies access	Recheck execution-time identity, active records and current source authority.

Kernel · Demo walkthrough

6 / 53

Next part of the demonstration

Execute one controlled run before adding recurrence.

CHAPTER 4

Demonstration · Run once and reconcile the producer result

Open	Who acts
Scheduler → selected published job → Run; Runs → selected run	Authorized runner and execution owner

A Scheduler acknowledgement is not the completed producer result. Inspect the durable run, attempts and business output.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Run reason / input override	Only offered typed values and the business reason	Do not bypass the reviewed schema with arbitrary payload.
Run / attempt ID	The returned durable identities	Retain these before repeating an interrupted request.
Producer result/reference	Vista, Pulse, source or registered-operation output	Proves what business work was actually completed.
State / failure code	Current queued, running, completed, failed or uncertain state	Use state-specific actions rather than an indiscriminate retry.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Open the published job and inspect owner, content and inputs.	The execution scope is understood.
2	Request one run with a reason and retain its ID.	The accepted work has a durable reference.
3	Open Runs and inspect queue/lease/attempt progress.	Admission and actual execution are distinguished.
4	Follow the producer reference to report/output/source result.	The owning module confirms its outcome.
5	Compare actual output and expected training values.	A successful technical invocation is not mistaken for correct business content.
6	Inspect audit and retained attempt history.	The actor, owner, definition and result can be traced together.

SCREEN 5

Run details, producer reference and audit verification

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

🔔

💬

🔗

Quick Links

SC

Studio Checker

Platform Demo Checker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 3

Job Definitions

Schedules

Job Runs

Pulse

Vista

Active Access Profile

Platform Demo Checker

Menus reflect this profile's permissions.

MediFlex > Core > Job Runs

View A B C D E Overview

Back

Core

Run 1 · Boot acceptance LETTER

Choose a valid Scheduler run audit file.

Status

Job version

Requested

Completed

Producer result reference

Succeeded

1

10/18/2026, 3:24:07 PM

10/18/2026, 3:24:27 PM

pulse-render://a1a76c43-92d9-4ec5-89fd-0e8438601b6d

Refresh Run

Download Audit

Audit file

Choose File

No file chosen

The file stays in your browser. Only its receipt and calculated hash are checked with the server.

Search loaded records...

Attempt

Status

Started

Dispatched

Completed

1

ACCEPTED

10/18/2026, 03:24:12 PM

10/18/2026, 03:24:16 PM

10/18/2026, 03:24:21 PM

Showing 1-1 of 1

<< < > >> Page 1 / 1

Search loaded records...

Sequence

Event

When

Reason

1

Queued

10/18/2026, 03:24:07 PM

2

Operator requested

10/18/2026, 03:24:07 PM

Isolated actual Boot execution acceptance

3

Claimed

10/18/2026, 03:24:12 PM

4

Dispatched

10/18/2026, 03:24:16 PM

5

Waiting external

10/18/2026, 03:24:21 PM

6

Succeeded

10/18/2026, 03:24:27 PM

- A successful scheduler run must have its producer outcome explained.
- Choose an actual exported audit file; the screenshot has no file selected.

Evidence to show before continuing

Show admission, run, attempt and actual producer output.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
The start response is lost	Find the existing run before requesting another; remote side effects may already exist.
Reconciliation required is shown	Inspect producer state and side effects before the permitted recovery action.

Kernel · Demo walkthrough

8 / 53

Next part of the demonstration

Create a governed schedule only after a controlled run behaves as intended.

CHAPTER 5

Demonstration · Create a calendar-aware one-time or recurring schedule

Open	Who acts
Scheduler → Schedules → New schedule; Business calendars	Schedule maker and independent reviewer

Declare local timing, timezone, calendar and missed-execution policy explicitly so operators can explain each occurrence.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Published job	The exact eligible definition	A schedule does not publish or authorize a draft job.
Trigger / local time / time zone	Choose the offered one-time or recurring form and an explicit zone	Compare the preview's local and UTC interpretation, especially around daylight-saving transitions.
Business calendar	The eligible published working calendar	Working days/exclusions constrain timing according to the selected policy.
Missed execution	Skip, fire once or bounded catch-up as offered	Catch-up limits 1–100 occurrences per batch in the inspected editor.
Grace / validity	The intended lateness allowance and schedule period	An expired occurrence should not be silently treated as current work.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Select the published job and enter the intended trigger.	The schedule has one explicit operation target.
2	Set zone and calendar, then preview future occurrences.	The displayed local/UTC times match the intended business timing.
3	Inspect a calendar exclusion and any relevant daylight-saving boundary.	Unexpected skips or duplicate-looking local times are identified before approval.
4	Choose missed-execution policy and small catch-up/grace bounds.	Outage recovery behavior is explicit.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
The next time is surprising	Compare timezone, calendar exclusion and missed-execution rule before changing the job.
The calendar changes	Review the resulting schedule impact and adopt the controlled version through the supported path.

Next part of the demonstration

Use backfill for a reviewed bounded past interval rather than recreating schedules to replay work.

CHAPTER 6

Demonstration · Choose retry, rerun, cancel, pause or retire correctly

Open	Who acts
Scheduler → selected job, schedule or run → eligible actions	Authorized operator

Recovery should preserve lineage and acknowledge that producer side effects may already have occurred.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Retry	An eligible attempt within the existing run lifecycle	Requires the registered operation's retry/idempotency policy.
Rerun	A linked new execution	Inspect the prior producer result before deliberately running again.
Cancel	A state-dependent request to stop work	Cancellation does not promise reversal of completed external effects.
Pause/resume schedule	Controls future admission	Already accepted runs have their own lifecycle.
Retire	Ends future use under the supported governed action	History remains; retirement is not physical deletion.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Inspect the failed/uncertain run and producer state first.	The operator knows whether work may already have happened.

Step	Demonstrator action	Point out on the screen
2	Use an automatic/eligible retry only under the published retry contract.	The attempt history remains under the original run.
3	If a new execution is intended, use Rerun with its reason.	The new run links to its predecessor.
4	Use Cancel only when offered and inspect its acknowledged result.	Requested cancellation and actual final state are distinct.
5	Pause a schedule to stop new admission, then resume after resolving the cause.	Existing runs are not silently erased.
6	Retire unused configuration through the supported action and inspect history.	Future use ends while audit evidence remains.

Evidence to show before continuing

Show the difference between original-run retry, linked rerun and paused future admission.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
The producer outcome is uncertain	Reconcile before retry/rerun; a timeout is not proof no external work occurred.
An action is disabled	Check current state, policy and permission rather than changing status directly.

Next part of the demonstration

Inspect the resulting producer/run lineage and use the audit verification view.

CHAPTER 7

Demonstration · Govern queue capacity and inspect worker leases

Open	Who acts
Scheduler → Queues; Workers; selected queue policy	Queue-policy maker/reviewer and runtime operator

Admission, queue concurrency and worker capacity jointly determine execution. A published job is not proof a healthy worker is available.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Queue policy	The eligible queue, capacity and concurrency settings	Small bounded values make impact easier to inspect.
Worker heartbeat / lease	Current worker liveness and claimed work	A visible worker row can be stale; inspect current heartbeat.
Queue pause / admission state	The permitted operational control	Pausing admission is different from reversing accepted work.
Impact preview / revision	The exact proposed queue-policy change	The reviewer needs the current affected workload.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Inspect queued/running work and worker heartbeat before changing capacity.	The actual bottleneck is identifiable.
2	Create the supported queue-policy revision and preview its impact.	The proposed change shows affected admission/capacity.
3	Complete the independent review and publication.	Runtime policy follows a controlled version.
4	Inspect a run moving from queued to leased/running.	Worker claim and producer execution are distinguishable.
5	Use the offered pause/resume/recovery operation and inspect retained leases.	Operational changes do not silently duplicate work.

SCREEN 8

Checker reviews queue-policy impact

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

Quick Links

Studio Checker

Platform Demo Checker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 5

Job Definitions

Schedules

Job Runs

Business Calendars

Queues & Workers

Pulse

Vista

Active Access Profile

Platform Demo Checker

Menus reflect this profile's permissions.

MediFlex > Core > Queues & Workers

View A B C D E Overview

Queue impact · Acceptance domain queue

Saved version 4. Counts reflect the current domain at preview time. Preview does not execute work.

Search loaded records...

Limit	Current Policy	Proposed Policy	Current Workload
Concurrent runs	4	4	0
Pending requests	1000	1000	0

Showing 1-2 of 2

Lower limits never cancel accepted work. New admissions wait until capacity is available. Retries retain their existing run identity.

Preview expires 10/18/2026, 7:41:34 PM. Return to the queue row to submit or review.

- Lower limits do not cancel already accepted work.
- Preview measures the current domain's workload at preview time.

Evidence to show before continuing

Show queue policy impact, worker heartbeat and a claimed run.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
Jobs remain queued	Check admission state, capacity, worker heartbeat and operation compatibility.
A worker disappears during work	Inspect lease expiry/reconciliation and producer outcome before re-admission.

Kernel · Demo walkthrough

14 / 53

Next part of the demonstration

Use bounded backfill or batch recovery once capacity and producer health are understood.

CHAPTER 8

Demonstration · Preview and admit only missing historical occurrences

Open	Who acts
Scheduler → Backfills → New plan	Backfill maker, independent reviewer and operator

Recover a bounded schedule gap without replacing existing success, failure or cancellation records.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Published schedule	The eligible schedule whose occurrences are missing	A new job definition alone has no historical occurrence contract.
Range start/end	A small prepared past interval; start included and end excluded	Inputs use browser timezone; compare the occurrence UTC/local preview.
Plan name / reason	Explain the gap and bounded recovery intent	The plan is reviewable before execution.
Occurrence action	Create missing or keep existing as actually previewed	Existing work must remain visible and unmodified.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Select the schedule and a small known training gap.	The proposed recovery scope is bounded.
2	Preview and inspect every occurrence and existing-run state.	Only genuinely missing occurrences are identified for admission.
3	Submit the exact plan for independent preview/review.	The reviewer sees current occurrence and input resolution.
4	Obtain a fresh approved-plan preview and Enqueue when eligible.	The operator admits only the reviewed missing work.
5	Follow the child runs and their producer results.	Backfill completion is backed by actual executions.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
All rows say Keep existing	That plan creates no replacements; choose an actual missing training interval if a new-run demonstration is required.
The preview became stale	Refresh and review the current plan rather than enqueueing outdated authority.

Next part of the demonstration

Inspect each recovered occurrence and retain plan/run/audit references.

CHAPTER 9

Demonstration · Run dependencies and resume from successful checkpoints

Open	Who acts
Scheduler → Batches → definition and selected execution	Batch maker/reviewer and operator

Express prerequisites across source refresh, reporting and communication so successful steps are retained during bounded recovery.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Steps / published jobs	Refresh source → report → dashboard/email/letter as appropriate	Each step references an eligible governed job.
Dependencies	The actual prerequisite graph	A predecessor must satisfy the configured success rule before dependent admission.
Execution owner/domain	A compatible authorized context for all steps	A batch does not add access to a child producer.
Checkpoint / child run	The recorded step result and linked execution	Checkpoint is step completion, not arbitrary internal producer resume.
Pause / stop admission / resume	The current permitted recovery action	These actions do not reverse completed business effects.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Build the named steps and explicit dependencies using published jobs.	The graph has a clear business sequence.
2	Preview inputs, owner authority and dependency validity.	Missing versions and invalid dependencies are resolved.
3	Save, independently review/publish and start the controlled batch.	Each admitted step receives its own run reference.
4	Inspect successful checkpoints and the first blocked/failed step.	The exact dependency preventing progress is identifiable.
5	On an isolated prepared run, Pause or Stop admission when permitted.	Accepted child jobs may finish while new admission is stopped.
6	Resolve the cause, Preview the current run and Resume when eligible.	Successful checkpoints remain retained instead of being blindly repeated.
7	Inspect every child producer before declaring final batch success.	A parent status is supported by the actual step outcomes.

Batch steps and prerequisites

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

🔔

💬

🔄

Quick Links

Studio Maker Platform Demo Maker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 7

Job Definitions

Schedules

Job Runs

Business Calendars

Queues & Workers

Backfills

Batches

Pulse

Vista

Nexora

Active Access Profile Platform Demo Maker

MediFlex > Core > Batches

View A B C D E Overview

Back

Core

Create batch

Save Draft

1 · Batch details

Name the batch and set a bound on its execution. Each job keeps its own retries, permissions and timeout.

Batch code

Batch name

Batch description

Concurrent steps

BROWSER_1792376991056

👤 Browser batch qbox 179237699

T Browser-authored dependency

1

Run deadline seconds

3600

2 · Steps and prerequisites

+ Add Step

Choose published jobs from this domain with the same execution identity. A step waits for all selected prerequisites to succeed. Leave prerequisites empty to start it immediately.

Step 1 · First report

Remove

Step name

Published job

Add prerequisite

👤 First report

✓ Boot acceptance VISTA · Version 1

✓ Choose a step

Starts when the batch is admitted.

Step 2 · Dependent report

Remove

Step name

Published job

Add prerequisite

👤 Dependent report

✓ Boot acceptance VISTA · Version 1

✓ Choose a step

Remove Prerequisite: First Report

- A step waits for all selected prerequisites to succeed.
- All selected published jobs must have compatible scope and execution identity.

Evidence to show before continuing

Show dependency graph, checkpoint, bounded pause/resume and retained successful work.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
A child has uncertain side effects	Reconcile that producer before resuming dependent work.
A checkpoint is missing	Inspect the actual child run/result; do not manually mark it successful.

Kernel · Demo walkthrough

19 / 53

Next part of the demonstration

Use run audit verification and retain batch, child run and producer references.

CHAPTER 10

Demonstration · Verify the complete automation trail

Open	Who acts
Scheduler → selected run → Audit / verification	Auditor or operator

Trace what was authorized, admitted, attempted and produced without confusing configuration review with business success.

Inputs for this part of the prepared example

Input shown on screen	Enter or select	Explain its significance
Job/schedule/batch version	The exact controlled definitions	Current drafts may differ from what executed.
Occurrence/run/attempt	Nominal timing, durable execution and tries	These identify different levels of the same automation trail.
Owner / input snapshot / producer	Execution authority and resolved values	Do not disclose private credential references unnecessarily in exported evidence.
Audit verification	The offered integrity/lineage result	A verified trail proves retained evidence integrity within its stated scope, not that all business content is correct.

Perform and explain

Step	Demonstrator action	Point out on the screen
1	Open one completed or failed training run and follow its definition and occurrence.	The authorized configuration can be identified.
2	Inspect owner, resolved inputs and all attempts.	Retries and failures remain attributable.
3	Follow the producer/output reference and any rerun/backfill/batch parent.	Cross-module lineage is complete for this example.
4	Run the available audit verification and inspect its result.	Any integrity/lineage concern remains explicit.
5	Retain the scoped references and actual result in UAT/support evidence.	The report does not overstate beyond the examined execution.

Evidence to show before continuing

Show definition → timing/admission → run/attempt → producer → recovery/audit lineage.

Demonstrate an exception when the prepared record supports it

Condition to explain	Supported recovery
Producer evidence is missing	Investigate the owning module rather than treating Scheduler completion as sufficient.
Verification reports a concern	Preserve the original records and escalate the exact failed boundary; do not rewrite audit rows.

Next part of the demonstration

Return to the job/schedule owner for a governed configuration change if needed.

CHAPTER 11

Additional demonstration · Access, prerequisites and example values

Prepare published source connection, Vista report/dashboard and Pulse email/letter templates with reviewed datasets. Use compatible domain and execution identity for batch jobs. Verify the physical worker/queue is available. Use isolated fixtures for failure, pause or capacity demonstrations; do not stop a shared production worker to create a failure.

Input / choice	Example or selection	Meaning / check
Operation	NEXUS_SOURCE_REFRESH_V1, VISTA_REPORT_V1, VISTA_DASHBOARD_V1, PULSE_EMAIL_V1 or PULSE_LETTER_V1	Choose the registered matching operation before selecting published content. A dataset alone is not automatically executable.
Run as / artifact	Approved execution identity; exact published content	The content selector supplies identity/version/revision/hash. Keep these pins through review and run inspection.
Input source	Value or eligible Dataset field binding	Dataset binding selects one unique record/value per binding; it is not implicit bulk execution.
Timeout / attempts / overlap	Example: 300 seconds, one attempt, Wait in queue	These are proposed demo values, not universal defaults. Multiple attempts require supported idempotent operation behavior.

Input / choice	Example or selection	Meaning / check
Schedule timezone / timing	Future UTC run-once instant; or 0 9 * * 1-5 in an intended IANA timezone	Compare local and UTC preview. Elapsed hourly timing and local 09:00 recurrence behave differently across DST.
Backfill range	A small interval with genuinely missing occurrences	Inspected bounds are 31 days and 100 occurrences; start included/end excluded. Existing failed runs are not missing.
Batch step / prerequisites	Published job pins; Refresh source before content jobs	Use compatible identity/domain and an acyclic dependency graph; inspected bounds are 32 jobs and 128 success dependencies.

Use assigned profiles for the stated responsibilities. If a menu is absent, inspect the selected domain/profile and ask the access administrator to check its mappings. Do not substitute a broad administrator profile. If an action is unavailable, check state, current revision and reviewer eligibility.

Data used in the procedures

The example is a service-review pack: a manager views work by region, receives an email summary for one case, and generates a formal letter for the same case. These are proposed walkthrough values, not records created in your Azure tenant.

Object	Example code / name
Connection	DEMO_SERVICE_API / Demo service records API
Source definition	DEMO_SERVICE_SOURCE / Service records
Primary dataset	DEMO_SERVICE_DATA / Service review dataset
Related-record dataset	DEMO_SERVICE_ITEMS / Service line items
Report	DEMO_SERVICE_REPORT / Service operations report
Dashboard	DEMO_SERVICE_DASH / Service operations dashboard
Email template	DEMO_SERVICE_EMAIL / Service review email
Letter template	DEMO_SERVICE_LETTER / Service review letter
Jobs	DEMO_REFRESH_JOB, DEMO_REPORT_JOB, DEMO_DASH_JOB, DEMO_EMAIL_JOB, DEMO_LETTER_JOB

record_id	recipient_name	region	status	amount
CASE1001	Alex Demo	North	OPEN	1200
CASE1002	Morgan Demo	South	COMPLETE	800
CASE1003	Taylor Demo	North	OPEN	500

record_id	recipient_name	region	status	amount
CASE1004	Casey Demo	West	REVIEW	1500

Also expose recipient_address, organisation_name, reference, correspondence_date and recipient_email. Keep amount numeric and correspondence_date in a supported date representation. The related item source has item_id, case_id, description and line_amount; case_id matches record_id.

The pack includes **DEMO_SERVICE_RECORDS.json** and **DEMO_SERVICE_ITEMS.json** as synthetic payload examples. They are not an import facility or a running API. Have the operator’s approved demo API serve them, or select an existing published business source with equivalent fields. The primary example totals four records and amount 4000 only when all four rows are in the authorized result; a limited preview or narrower profile may show less.

CHAPTER 12

Additional demonstration · Define, publish and run a job

Use this route when you need to define, publish and run a job.

- 1. [Published artifact and Scheduler operation bindings](#)
- 2. [Create and govern a job definition](#)
- 3. [Configure report, dashboard, email and letter operations](#)
- 4. [Bind current dataset values to job inputs](#)
- 5. [Execute a job and inspect its result](#)
- 6. [Inspect and verify execution audit evidence](#)

CHAPTER 13

Additional demonstration · Step 1 · Published artifact and Scheduler operation bindings

Before you act	Required context
Screen / entry	Job Definitions and start a new draft only after the intended business artifact is published.
Who acts	Authorized operator / eligible independent reviewer as specified below

Before you act	Required context
You need	Prepare published source connection, Vista report/dashboard and Pulse email/letter templates with reviewed datasets. Use compatible domain and execution identity for batch jobs. Verify the physical worker/queue is available. Use isolated fixtures for failure, pause or capacity demonstrations; do not stop a shared production worker to create a failure.

Steps and field guidance

1. Open Job Definitions and start a new draft only after the intended business artifact is published.
2. Select the registered operation that matches the artifact and output in the table below.
3. Confirm the Run as identity can access that artifact and its source data. Continue to the job-definition procedure to save, preview and govern the exact binding.

Content / purpose	Registered business operation	Produced result
Source connection	NEXUS_SOURCE_REFRESH_V1 · Refresh published source connection	New authorized source snapshot
Vista report	VISTA_REPORT_V1 · Run published Vista report	Published report output
Vista dashboard	VISTA_DASHBOARD_V1 · Run published Vista dashboard (registered operation version 2)	Dashboard snapshot with all report tiles checked
Pulse email	PULSE_EMAIL_V1 · Generate published Pulse email	Generated email file; sending is separate
Pulse letter	PULSE_LETTER_V1 · Generate published Pulse letter	Rendered letter/PDF output
Private draft maintenance	VISTA_RECOVERY_CLEANUP_V1 / PULSE_RECOVERY_CLEANUP_V1	Bounded removal of expired recovery copies; maintenance permission required

Published artifact + authorized Run as identity + inputs → reviewed Job Definition → published version → Schedule / Run now / Batch step → worker → owning producer → result reference

A dataset is consumed by a report/template or used for job input binding. It is not automatically a standalone executable job. The source refresh operation refreshes a published connection’s snapshot. Additional external business operations must be registered with a supported input/result contract and permission model before they appear in the selector.

Expected result

The operator can identify which owning module produces the result and which job/schedule/run record coordinates it.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 14

Additional demonstration · Step 2 · Create and govern a job definition

Before you act	Required context
Screen / entry	Scheduler → Job Definitions → New job
Who acts	Scheduler Maker
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Scheduler Maker

Open: Scheduler → Job Definitions → New job

1. Choose Business operation: Run published Vista report. Select the approved Run as execution identity.
2. Enter Job name: Service report generation and Job code: DEMO_REPORT_JOB. Add a useful description.
3. In Published content choose Service operations report and its published version by name. The application fills its identity, version, revision and content hash.
4. Add Reason as a text Value: Reference-data setup: generate the approved service operations report. Use the offered contract inputs; do not invent reportId/version values.
5. Set Timeout to 300 seconds for the walkthrough if appropriate. Start with Maximum attempts 1 and overlap policy Wait in queue. Verify retry tuning separately.
6. Save draft. Select Preview on the saved job row. Inspect the operation, execution context, resolved values and pinned artifact.
7. Return to the row and Submit for review. Checker independently previews the same revision, then Approve or Return. Publish the approved job version.

SCREEN 1

Job inputs bind a named published report

Existing application recorded test · example records may differ from your tenant · click to enlarge

MediFlex
Operations Workspace

🏠 Your Workspace

🔍 Find a screen...

📅 Business journeys

📊 My Dashboard

🔍 Core

📅 Scheduler 3

📅 Job Definitions

📅 Schedules

📅 Job Runs

📧 Pulse

📊 Vista

📅 Nexora

Active Access Profile

Platform Demo Maker

Menus reflect this profile's permissions.

🔍 Search screens and work areas...

Ctrl K

🔗 Quick Links

Studio Maker
Platform Demo Maker

Job name

Browser scheduled artifact qbox 1792317514

Job code

BROWSER_QBOX_1792317514167

Description

Enter description

Published content

Published report

Karthi_Vista_Test · KARTHI_VISTA_TEST · v1

This job uses version 1 of Karthi_Vista_Test. A later publication requires a new job revision and review.

Inputs

Use a value or select one record from a published dataset. Dataset values are refreshed and validated when the job runs.

🔍 Search loaded records...

Input	Source	Value Or Field	Selected Record	Actions
Reason	Value	Scheduled reporting browser acceptance		

Showing 1-1 of 1

«

<

>

»

Page 1 / 1

Input name

Choose an input

Input source

Value

Value type

TEXT

Value

+ Add Input

Execution policy

Timeout (seconds)

300

Maximum attempts

1

Initial retry delay (seconds)

30

Maximum retry delay (seconds)

300

If a previous run is still active

Wait in queue

The registered operation supports a stable retry key.

- The selector supplies the version and content hash automatically.
- Execution policy is saved with the job definition.

Kernel · Demo walkthrough

26 / 53

SCREEN 2

Job preview resolves inputs without executing the business operation

Existing application recorded test · example records may differ from your tenant · click to enlarge

The screenshot displays the MediFlex Operations Workspace interface. The top navigation bar includes the Kernel Health logo, the MediFlex Operations Workspace title, a search bar, and various utility icons. The left sidebar contains a 'Your Workspace' section with links to Business journeys, My Dashboard, Core, Scheduler, Job Definitions (highlighted), Schedules, and Job Runs. Below this is an 'Active Access Profile' section for 'Platform Demo Maker'. The main content area shows a breadcrumb trail 'MediFlex > Core > Job Definitions' and a title bar 'Preview · Browser scheduled artifact qbox 1792317514110'. A message states: 'Current input values and the registered operation are shown. Preview does not execute the business operation.' Below this, a table provides details: 'Saved version' (1 · Revision 1), 'Preview valid until' (10/18/2026, 3:43:45 PM), and 'Timeout' (300 seconds). A search bar for 'Search loaded records...' is present. A table lists input values:

Input	Current Value
reason	Scheduled reporting browser acceptance
reportId	becfa741-93cf-4fbe-a2a9-455f6321af70
versionId	22507ced-4e69-46c2-a9be-c28793a9bef6
contentHash	80e46d065ba9d05c6e14752dcdbcb692c75540a2a26ae0092c8ebbdd5a5496f9
expectedRevision	4

At the bottom of the table, it says 'Showing 1-5 of 5'. Below the table, a message reads: 'Return to the job row to submit or review. Changed data, permissions or draft details require a new preview.'

- Preview establishes current saved-definition eligibility.
- It does not generate the report; an actual run comes later.

Expected result

A published report job is available to Run a job, Schedules and compatible batch steps.

A job's saved artifact reference is pinned. Publishing report version 2 does not make this job use version 2. Change the job through its supported new-version/review/publication flow and then update dependent schedules/batches explicitly.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 15

Additional demonstration · Step 3 · Configure report, dashboard, email and letter operations

Before you act	Required context
Screen / entry	Job Definitions → New job for each artifact
Who acts	Scheduler Maker → Checker → Publisher
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Scheduler Maker → Checker → Publisher
Open: Job Definitions → New job for each artifact

Job	Business operation / content	Inputs to verify
DEMO_DASH_JOB	Run published Vista dashboard → Service operations dashboard	Reason; dashboard version/revision/hash supplied by the selector
DEMO_EMAIL_JOB	Generate published Pulse email → Service review email	Record key = CASE1001 when bound; sender and recipient; reason and any offered merge inputs
DEMO_LETTER_JOB	Generate published Pulse letter → Service review letter	Record key = CASE1001 when bound; reason and any offered merge inputs; no email sender/recipient
DEMO_REFRESH_JOB	Refresh published source connection → Demo service records API	Reason; connection version/revision/hash supplied by the selector

1. For each job, select the matching registered operation before choosing its published content. Email and letter choices are filtered by channel.
2. Use the same approved execution identity and domain for all jobs if they will form one batch. Verify this identity's access to every dataset and report behind the artifacts.
3. For the email job use a controlled sender and recipient. Example-only addresses such as sender@example.test and recipient@example.test are suitable for explaining file composition; they are not deliverable mailboxes.
4. For both bound Pulse templates supply the exact parent Record key. The producer loads the saved dataset mapping and current authorized record at execution.
5. Save, Preview, Submit for review, obtain an independent current preview and decision, then Publish each job. Record its published version.

SCREEN 3

Dashboard job selects a published dashboard

Existing application recorded test · example records may differ from your tenant · click to enlarge

MediFlex
Operations Workspace

Search screens and work areas...

Ctrl K

Quick Links

Studio Maker
Platform Demo Maker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 3

Job Definitions

Schedules

Job Runs

Pulse

Vista

Nexora

Active Access Profile
Platform Demo Maker
Menus reflect this profile's permissions.

Published content

Published dashboard

Analytical dashboard verification · ANALYTICS_DASH_900C90EC73 · v1

This job uses version 1 of Analytical dashboard verification. A later publication requires a new job revision and review.

Every linked report is checked when the dashboard runs. All tiles must be available for the job to complete successfully.

Inputs

Use a value or select one record from a published dataset. Dataset values are refreshed when the job runs.

Search loaded records...

Input	Source	Value Or Field	Selected Record	Actions
reason	Value	Scheduled reporting browser acceptance		

Showing 1-1 of 1

<< < > >> Page 1 / 1

Input name

Choose an input

Input source

Value

Value type

TEXT

Value

Enter value

+ Add Input

Execution policy

Timeout (seconds)

300

Maximum attempts

1

Initial retry delay (seconds)

30

Maximum retry delay (seconds)

300

If a previous run is still active

Wait in queue

The registered operation supports a stable retry key.

- Every pinned report tile is checked at execution.
- All tiles must be available for a successful scheduled dashboard result.

Kernel · Demo walkthrough

29 / 53

SCREEN 4

Source-refresh job selects a published connection

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

🔔

💬

🔄

Quick Links

SM

Studio Maker

Platform Demo Maker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 3

Job Definitions

Schedules

Job Runs

Pulse

Vista

Nexora

Active Access Profile

Platform Demo Maker

Menus reflect this profile's permissions.

Published content

Published source connection

External test records · NATIVE_CONNECTION_F8170595C9 · revision 6

This job uses revision 6 of External test records. A later publication requires a new job revision and review.

Nexus refreshes the published source using its saved authentication reference and current execution permissions. Dataset reads use the resulting authorized snapshot.

Inputs

Use a value or select one record from a published dataset. Dataset values are refreshed when the job runs.

Search loaded records...

Input	Source	Value Or Field	Selected Record	Actions
reason	Value	Scheduled reporting browser acceptance		

Showing 1-1 of 1

Page 1 / 1

Input name

Choose an input

Input source

Value

Value type

TEXT

Value

Enter value

+ Add Input

Execution policy

Timeout (seconds)

300

Maximum attempts

1

Initial retry delay (seconds)

30

Maximum retry delay (seconds)

300

If a previous run is still active

Wait in queue

The registered operation supports a stable retry key.

- Refresh uses the saved authentication reference and current execution permissions.
- Subsequent dataset reads consume the authorized snapshot.

Expected result

Four business-content jobs and the optional source-refresh job are published. Each points to the intended approved artifact version.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 16

Kernel · Demo walkthrough

30 / 53

Additional demonstration · Step 4 · Bind current dataset values to job inputs

Before you act	Required context
Screen / entry	Job definition → Inputs
Who acts	Scheduler Maker
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Scheduler Maker

Open: Job definition → Inputs

1. Choose an input permitted by the selected operation's schema. For example, bind the email job's recipient to the authorized primary dataset's recipient_email.
2. Set Input source to Dataset field. Select the published dataset, value field recipient_email, unique key field record_id and exact record CASE1001.
3. Add the input and inspect the resulting input grid. Keep Record key consistent with the Pulse template's intended parent record.
4. Save and Preview. Confirm the currently resolved value and its dataset version/contract. Submit/review/publish the changed definition as required.
5. At execution the binding is read and validated again. Verify a changed value only using a controlled source update followed by the required snapshot refresh; then compare the actual output.

Expected result

The job reads a current authorized value for one explicitly selected record. Missing access, missing/ambiguous keys or incompatible values fail rather than silently using an unrelated record.

Dataset field binding is not "run once per every dataset row." The inspected input model selects one record per binding. Do not claim bulk-recipient fan-out, arbitrary output-to-input piping or a generic ETL loop. Use a registered business operation for any additional bulk behavior.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

Additional demonstration · Step 5 · Execute a job and inspect its result

Before you act	Required context
Screen / entry	Scheduler → Job Runs → Run a job
Who acts	Authorized Scheduler operator
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Authorized Scheduler operator

Open: Scheduler → Job Runs → Run a job

1. Select the published Service report generation job. Preview current inputs, provide the execution reason and select Queue run. Record the run reference.
2. Refresh the run. Follow Queued → Claimed/Running → Waiting external where applicable → Succeeded or an explicit failure.
3. Open Details. Inspect the exact job version, attempt information, execution timestamps and Producer result reference.
4. Repeat for the dashboard, email and letter jobs, one at a time for the user walkthrough.
5. For each producer result, open the appropriate authorized Vista/Pulse output screen or access-controlled result view. Compare it with the manual output and intended case.
6. Record actual outcomes in LIVE_DEMO_CHECKLIST.csv. Do not mark a queued/accepted job as a successful generated output.

Job	Required success evidence	Extra check
Report	Scheduler SUCCEEDED + completed report result reference	Expected dataset scope, values and published report version
Dashboard	Scheduler SUCCEEDED + completed snapshot	Every tile available; PARTIAL is not success
Email	Scheduler SUCCEEDED + READY generated-email reference	Correct record, sender/recipient in file; no claim of send
Letter	Scheduler SUCCEEDED + READY letter/PDF reference	Correct recipient/reference and pagination

Scheduled work can run as a service identity different from the operator. Producer output lists are access-controlled and may be requester-scoped. A result absent from the operator's personal Generated Emails/Rendered letters list is not proof that no output exists. Use the authorized result access path and the scheduler's recorded producer reference; do not broaden permissions just to expose another identity's output.

Expected result

All four actual executions have a result reference and a verified output. Rehearse these before enabling a recurring schedule.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 18

Additional demonstration · Step 6 · Inspect and verify execution audit evidence

Before you act	Required context
Screen / entry	Scheduler → Job Runs → Details
Who acts	Authorized auditor/operator
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Authorized auditor/operator

Open: Scheduler → Job Runs → Details

1. Open a completed run and identify job version, requested/completed timestamps, status and producer result reference.
2. Review attempts and ordered audit events such as queued, claimed, dispatched, waiting external and succeeded/failed.
3. Select Download audit and retain the exported file unchanged.
4. Use the Audit file / Verify audit file control to select that file. The browser calculates its hash and verifies its receipt with the server; the full file remains in the browser.

- 5. If using tamper detection, use a separate edited copy of a demo audit export. Its verification should fail without altering the original server history.
- 6. Open the job/schedule/queue/backfill/batch Audit history where offered to connect the approval and operational actions to execution.

SCREEN 5

Run details, producer reference and audit verification

Existing application recorded test · example records may differ from your tenant · click to enlarge

KERNEL HEALTH

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

Quick Links

Studio Checker

Platform Demo Checker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 3

Job Definitions

Schedules

Job Runs

Pulse

Vista

Active Access Profile

Platform Demo Checker

Menus reflect this profile's permissions.

MediFlex

Core

Job Runs

View A B C D E Overview

Run 1 · Boot acceptance LETTER

Choose a valid Scheduler run audit file.

Status

Job version

Requested

Completed

Producer result reference

Succeeded

1

10/18/2026, 3:24:07 PM

10/18/2026, 3:24:27 PM

pulse-render://a1a76c43-92d9-4ec5-89fd-0e8438601b6d

Refresh Run

Download Audit

Audit file

Choose File

No file chosen

The file stays in your browser. Only its receipt and calculated hash are checked with the server.

Search loaded records...

Attempt

Status

Started

Dispatched

Completed

1

ACCEPTED

10/18/2026, 03:24:12 PM

10/18/2026, 03:24:16 PM

10/18/2026, 03:24:21 PM

Showing 1-1 of 1

Page 1 / 1

Search loaded records...

Sequence

Event

When

Reason

1

Queued

10/18/2026, 03:24:07 PM

2

Operator requested

10/18/2026, 03:24:07 PM

Isolated actual Boot execution acceptance

3

Claimed

10/18/2026, 03:24:12 PM

4

Dispatched

10/18/2026, 03:24:16 PM

5

Waiting external

10/18/2026, 03:24:21 PM

6

Succeeded

10/18/2026, 03:24:27 PM

- A successful scheduler run must have its producer outcome explained.
- Choose an actual exported audit file; the screenshot has no file selected.

Expected result

The authorized user can trace who created/reviewed/published the definition, what version ran, what happened at each attempt and how the output reference is preserved.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

Kernel · Demo walkthrough

34 / 53

Additional demonstration · Create and operate a schedule

Use this route when you need to create and operate a schedule.

- 1. Business calendars, time zones and DST
- 2. Create a one-time or recurring schedule
- 3. Timeouts, retries, overlaps and misfires
- 4. Queues and workers: govern capacity and execution
- 5. Rerun, cancel, pause, resume and retire

Additional demonstration · Step 1 · Business calendars, time zones and DST

Before you act	Required context
Screen / entry	Scheduler → Business Calendars → New calendar
Who acts	Calendar Maker → Checker → Publisher
You need	Prepare published source connection, Vista report/dashboard and Pulse email/letter templates with reviewed datasets. Use compatible domain and execution identity for batch jobs. Verify the physical worker/queue is available. Use isolated fixtures for failure, pause or capacity demonstrations; do not stop a shared production worker to create a failure.

Steps and field guidance

Who: Calendar Maker → Checker → Publisher

Open: Scheduler → Business Calendars → New calendar

- 1. Enter a calendar name/code and choose the intended IANA timezone, for example America/New_York for a US business-day walkthrough.
- 2. Select the working days and local working windows, for example Monday–Friday 09:00–17:00. The end of a window is excluded.
- 3. Add a date exception: either closed for a holiday or explicit special working hours. Note that this replaces the usual working day.
- 4. Save draft and Preview the dates. Checker previews independently, approves/returns, and an authorized publisher publishes.

5. Select this published calendar in a new schedule. Review the upcoming occurrences in both the calendar's local timezone and UTC.
6. For cron schedules choose a DST gap policy (skip or shift forward) and fold policy (first, second or both occurrences). Preview the relevant boundary dates before enabling.

SCREEN 6

Business calendar authoring

Existing application recorded test · example records may differ from your tenant · click to enlarge

MediFlex Operations Workspace

Search screens and work areas... Ctrl K

Quick Links

Studio Maker Platform Demo Maker

MediFlex > Core > Business Calendars

View A B C D E Overview

← Back Core **Create calendar** Save Draft

Calendar details

Calendar name: Enter calendar name Calendar code: Enter calendar code Description: Enter description

Working week

Time zone: UTC

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Schedules use this calendar's time zone. Date exceptions below replace the usual working day.

Working windows

Use 24-hour local times. The end is excluded; 24:00 means midnight at the end of the day.

Window start: 09:00 Window end: 17:00 Remove Window 1

+ Add Working Window

Holidays and special dates

Close a holiday or set special working hours, including a normally closed weekend.

+ Add Date Exception

- Working windows use local 24-hour times.
- Holidays and special hours are versioned and reviewed before publication.

Expected result

The schedule respects a specific published calendar version. Local-time expectations are checked against its actual preview.

Fixed elapsed intervals do not behave like “09:00 local every day” across daylight-saving changes. Editing/publishing a calendar does not automatically rebind existing schedules to a new calendar version.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 21

Additional demonstration · Step 2 · Create a one-time or recurring schedule

Before you act	Required context
Screen / entry	Scheduler → Schedules → New schedule
Who acts	Scheduler Maker
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Scheduler Maker

Open: Scheduler → Schedules → New schedule

1. Select the published job version. Enter name/code, for example Service report weekday schedule / DEMO_REPORT_SCHEDULE.
2. For the first live execution choose Run once and a future execution time with enough time to complete review. Enter the UTC instant in the field explicitly labelled UTC.
3. Choose the display/calendar Time zone. Inspect both UTC and local time in Preview rather than relying on the browser’s timezone.
4. For repetition choose Fixed elapsed interval with a UTC anchor and interval seconds, or Calendar schedule with a five-field cron expression.
5. Optionally select a published Business calendar. It pins that calendar version and constrains the schedule to its allowed dates/windows.
6. Choose the missed-execution policy and limits. Save draft, Preview upcoming dates and current input bindings, then Submit for review.
7. Checker performs its own current Preview and Approve or Return. An authorized publisher uses Publish and enable when the approved revision has a fresh preview.
8. Open Job Runs after the occurrence. Verify the schedule’s run and producer result, then pause or retire the demo schedule after use.

SCREEN 7

Schedule timing, calendar and missed-execution behavior

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel Health

MediFlex

Operations Workspace

Search screens and work areas...

Ctrl K

Quick Links

Studio Maker Platform Demo Maker

Your Workspace

Find a screen...

Business journeys

My Dashboard

Core

Scheduler 7

Job Definitions

Schedules

Job Runs

Business Calendars

Queues & Workers

Backfills

Batches

Pulse

Vista

Nexora

Active Access Profile Platform Demo Maker

MediFlex > Core > Schedules

View A B C D E Overview

Back

Core

Create schedule

Save Draft

Job and schedule details

Published job version

Schedule name

Schedule code

Description

The schedule keeps this exact job version. Saving a draft does not enable execution.

Business calendar

Published business calendar

Optionally limit this schedule to approved working hours and holidays. The schedule keeps the selected calendar version.

Timing

Trigger type

Time zone

Execution date and time (UTC)

Dates above use UTC. Preview also shows them in the selected time zone. Fixed intervals measure elapsed seconds and do not shift with daylight saving.

Missed executions

When the scheduler resumes after downtime

Maximum occurrences per batch

Late execution grace (seconds)

Choose the execution date and time in UTC.

- A schedule pins an exact published job version.
- Saving a draft does not enable execution.

Trigger	Example	Use
Run once	A chosen future UTC instant	One scheduled execution
Fixed elapsed interval	Anchor in UTC; every 3600 seconds	Elapsed hourly cadence independent of local DST
Calendar schedule	0 9 * * 1-5 in the chosen timezone	09:00 on weekdays
Calendar schedule	0 9 * * 1	Monday at 09:00
Calendar schedule	0 9 1 * *	First day of each month at 09:00

Create separate schedules for the email, letter and dashboard jobs using the same procedure when they need independent timing. A schedule is a job trigger; the inspected schedule selector does not establish timer-triggered batch execution. Verify batch execution through Batches, as described later.

Kernel · Demo walkthrough

38 / 53

Check before continuing: The saved reviewed schedule references the intended published job version, timezone and calendar. Its upcoming occurrences and missed-execution behavior match the requested timing.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 22

Additional demonstration · Step 3 · Timeouts, retries, overlaps and misfires

Before you act	Required context
Screen / entry	Timeouts, retries, overlaps and misfires
Who acts	Authorized operator / reviewer
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Setting	Where	Meaning / user walkthrough
Timeout	Job Definition	Maximum operation time; a timeout is not evidence that a remote side effect never happened
Maximum attempts / retry delay	Job Definition	Bounded retries only when the registered operation supports the required stable retry key
Wait in queue	Job overlap policy	A later occurrence waits while the previous run is active
Skip overlapping work	Job overlap policy	Reject/skip overlapping execution according to the published job policy
Bounded concurrency	Job overlap policy	Permit a limited number of concurrent runs; queue-wide limits still apply
Skip expired occurrences	Schedule missed-execution policy	Do not replay expired occurrences
Fire once	Schedule missed-execution policy	Coalesce missed work into one occurrence according to the scheduler policy
Catch up	Schedule missed-execution policy	Admit bounded missed occurrences; configure per-batch limit and grace

The inspected job editor bounds timeout to 1–86400 seconds, attempts to 1–10 for eligible operations, and bounded concurrency to 1–32. Schedule catch-up batches are limited to 1–100 occurrences. Use sensible small demo values instead of treating these maxima as recommendations.

1. Show the published policy on one job and schedule. Explain its effect before changing it.
2. Use a prepared isolated failing operation/run to show an allowed retry and its retained attempt history. Do not deliberately stop a shared production worker.
3. Compare an automatic retry with a manual Rerun. A retry belongs to the existing run/attempt lifecycle; rerun creates a linked new execution.
4. If the state is Reconciliation required, inspect producer status and possible side effects before any permitted recovery.

Expected result
Failure handling is bounded, auditable and consistent with the registered producer contract. Scheduling does not make every operation safely repeatable.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 23

Additional demonstration · Step 4 · Queues and workers: govern capacity and execution

Before you act	Required context
Screen / entry	Scheduler → Queues & Workers
Who acts	Queue Maker/Checker; authorized runtime operator
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Queue Maker/Checker; authorized runtime operator
Open: Scheduler → Queues & Workers

1. Inspect the existing queue policy for the current demo domain. Only use a dedicated demo queue for a live capacity-change walkthrough.
2. Create or edit a queue-policy draft with name/code, maximum concurrent runs and maximum pending requests.

3. Preview impact. Compare Current policy, Proposed policy and Current workload. Submit for review; the Checker previews independently and approves/returns; publish when authorized.
4. Verify Pause claims and Resume claims only on the isolated queue. Pausing claims does not cancel accepted work; pending capacity limits still apply.
5. Open Workers. Show worker state, last heartbeat, concurrency and active leases. The inspected UI describes 30-second reports and stale status after two minutes.
6. Explain Drain worker: accepted work continues while new claims stop. Resume when eligible. Retire a worker record only when its current lifecycle allows it.
7. Open Audit history for the relevant policy and worker actions.

SCREEN 8

Checker reviews queue-policy impact

Existing application recorded test · example records may differ from your tenant · click to enlarge

Kernel HEALTH MediFlex Operations Workspace

Search screens and work areas... Ctrl K

Quick Links

Studio Checker Platform Demo Checker

MediFlex > Core > Queues & Workers

View A B C D E Overview

← Back Core Queue impact · Acceptance domain queue

Saved version 4. Counts reflect the current domain at preview time. Preview does not execute work.

Search loaded records...

Limit	Current Policy	Proposed Policy	Current Workload
Concurrent runs	4	4	0
Pending requests	1000	1000	0

Showing 1-2 of 2

Lower limits never cancel accepted work. New admissions wait until capacity is available. Retries retain their existing run identity.

Preview expires 10/18/2026, 7:41:34 PM. Return to the queue row to submit or review.

- Lower limits do not cancel already accepted work.
- Preview measures the current domain's workload at preview time.

Expected result

The run shows the difference between a reviewed capacity policy, an operational pause and a physical running worker.

The UI manages policy and registered worker state; it does not install a new physical worker process. Do not drain or pause a shared production worker simply to create an on-screen failure for this example.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 24

Additional demonstration · Step 5 · Rerun, cancel, pause, resume and retire

Before you act	Required context
Screen / entry	Job Runs, Schedules and Job Definitions → eligible row actions
Who acts	Authorized operator
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

1. Open the existing run, schedule or job and inspect its current status and last producer result.
2. Choose the single action that matches your need from the table below. Check the effect on already accepted work before confirming.
3. For a rerun, preview current inputs and supply the reason; for schedule resumption, inspect upcoming dates and missed-execution behavior.
4. Confirm the eligible action once. Refresh the originating record, retain the new receipt/reference and compare it with the original history.

Who: Authorized operator

Open: Job Runs, Schedules and Job Definitions → eligible row actions

Action	Procedure	Important effect
Rerun a completed/failed eligible run	Open run → Rerun → Preview current inputs → reason → Queue run	New linked run; same approved job version; current data may differ

Action	Procedure	Important effect
Cancel queued run	Select Cancel queued run and confirm the reason	Only eligible queued work; not a rollback of accepted external work
Pause schedule	Open eligible schedule row → Pause → reason	Stops future scheduling; already accepted/queued runs remain separate
Resume schedule	Preview upcoming dates/current inputs, then Resume when eligible	Missed-execution policy governs the resumption
Retire schedule	Use its offered retirement action and reason	Retains historical occurrences/runs
Retire job	Use the job's permitted retirement action	Stops new runs for the retired job; accepted work/history are retained
Create new job version	Branch/edit the supported draft → save → preview → review → publish	Dependents must explicitly select the new published version

Do not rerun uncertain external effects merely because the screen timed out. Reconciliation-required runs block automatic rerun. Changing a record to inactive or retired is not a universal cancellation or deletion mechanism.

Expected result

Every operator action has a reason and a retained audit trail. The original failed run is still visible after recovery.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 25

Additional demonstration · Recover missed work and run dependent batches

Use this route when you need to recover missed work and run dependent batches.

1. Backfill missing schedule occurrences
2. Batch dependencies across source refresh, Vista and Pulse
3. Pause and resume a batch using checkpoints
4. Update, deactivate and retain historical records

CHAPTER 26

Additional demonstration · Step 1 · Backfill missing schedule occurrences

Before you act	Required context
Screen / entry	Scheduler → Backfills
Who acts	Backfill Maker → Checker → operator
You need	Prepare published source connection, Vista report/dashboard and Pulse email/letter templates with reviewed datasets. Use compatible domain and execution identity for batch jobs. Verify the physical worker/queue is available. Use isolated fixtures for failure, pause or capacity demonstrations; do not stop a shared production worker to create a failure.

Steps and field guidance

Who: Backfill Maker → Checker → operator

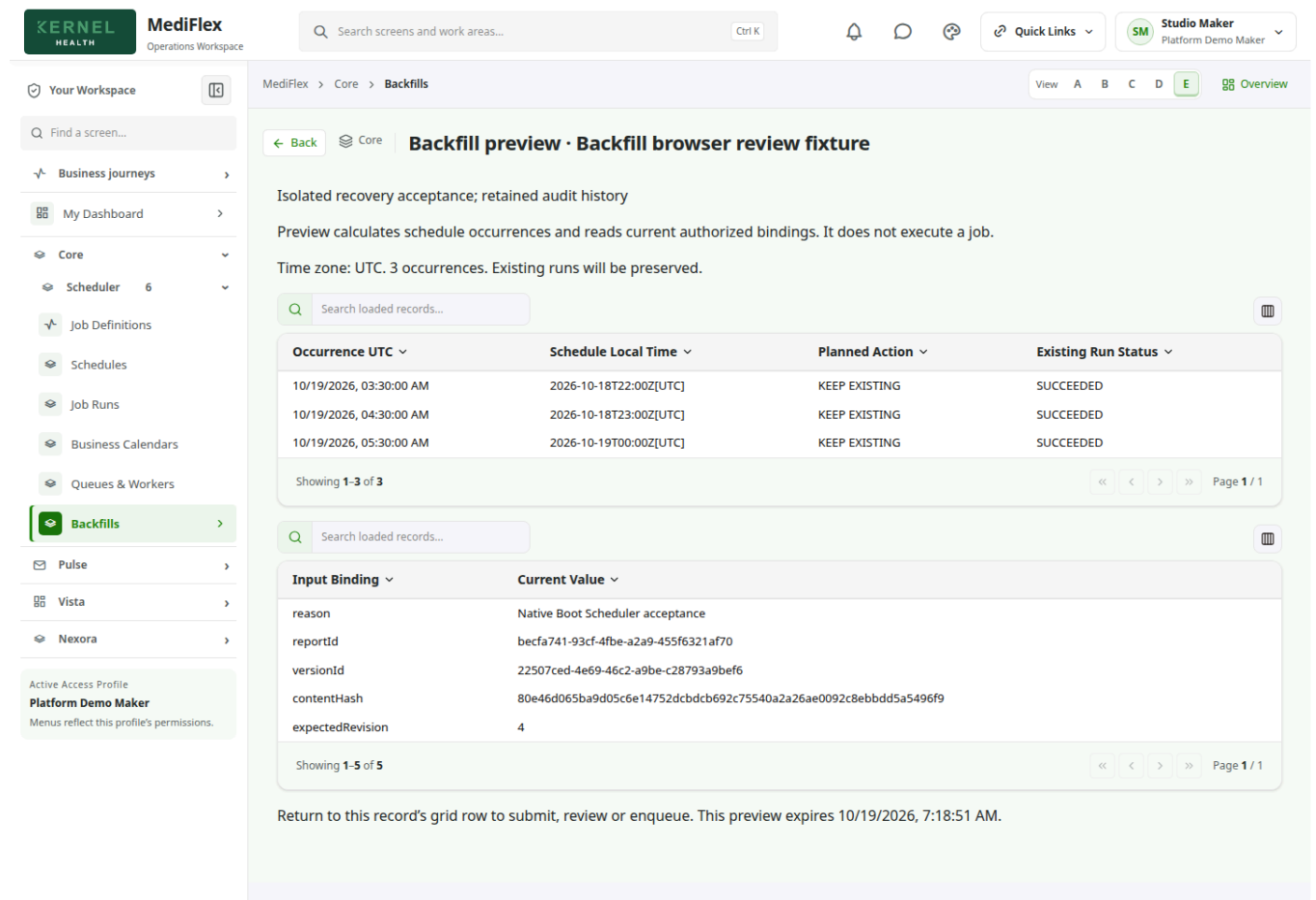
Open: Scheduler → Backfills

1. Choose a published schedule and a small prepared past time range with known missed occurrences. Range start/end use your browser timezone; the start is included and the end is excluded. Name the plan and explain the business reason.
2. Preview the plan. Inspect each occurrence, its timezone/UTC interpretation, planned action and any existing run status. Preview does not execute the job.
3. Confirm that existing runs are preserved and that only missing occurrences should be admitted. Check current input values and scope.
4. Submit for review. Checker previews the same plan independently and approves or returns.
5. Obtain a fresh approved-plan preview and select Enqueue. Follow the admitted runs and their producer results.
6. Use Pause/Resume only when eligible. Use Retire backfill to stop further admission when that lifecycle action is allowed. Existing accepted runs continue separately. Inspect progress and the audit history.

SCREEN 9

Backfill preview preserves existing runs

Existing application recorded test · example records may differ from your tenant · click to enlarge



MediFlex Operations Workspace

Search screens and work areas... Ctrl K

Quick Links Studio Maker Platform Demo Maker

MediFlex > Core > Backfills

View A B C D E Overview

← Back Core **Backfill preview · Backfill browser review fixture**

Isolated recovery acceptance; retained audit history

Preview calculates schedule occurrences and reads current authorized bindings. It does not execute a job.

Time zone: UTC. 3 occurrences. Existing runs will be preserved.

Search loaded records...

Occurrence UTC	Schedule Local Time	Planned Action	Existing Run Status
10/19/2026, 03:30:00 AM	2026-10-18T22:00Z[UTC]	KEEP EXISTING	SUCCEEDED
10/19/2026, 04:30:00 AM	2026-10-18T23:00Z[UTC]	KEEP EXISTING	SUCCEEDED
10/19/2026, 05:30:00 AM	2026-10-19T00:00Z[UTC]	KEEP EXISTING	SUCCEEDED

Showing 1-3 of 3

Search loaded records...

Input Binding	Current Value
reason	Native Boot Scheduler acceptance
reportId	becfa741-93cf-4fbc-a2a9-455f6321af70
versionId	22507ced-4e69-46c2-a9be-c28793a9bef6
contentHash	80e46d065ba9d05c6e14752dcdbdc692c75540a2a26ae0092c8ebbdd5a5496f9
expectedRevision	4

Showing 1-5 of 5

Return to this record's grid row to submit, review or enqueue. This preview expires 10/19/2026, 7:18:51 AM.

- This historical example shows KEEP EXISTING for all listed occurrences; it would not create replacements for them.
- Use the actual UTC/local preview in your environment and resolve any discrepancy before enqueueing.

Expected result

Only missing occurrences are queued under the approved bounded plan; existing success, failure and cancellation history is preserved.

The inspected backfill is limited to a 31-day window and 100 planned occurrences. It uses current authorized input data, not a historical as-of reconstruction. A failed existing run is not “missing”; use its eligible rerun/recovery path.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

Additional demonstration · Step 2 · Batch dependencies across source refresh, Vista and Pulse

Before you act	Required context
Screen / entry	Scheduler → Batches → Batch definitions → New batch
Who acts	Batch Maker
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Batch Maker

Open: Scheduler → Batches → Batch definitions → New batch

1. Enter DEMO_SERVICE_BATCH / Service review pack. Choose a small concurrent-step limit and a reasonable run deadline.
2. Add the published source-refresh job as step Refresh source, with no prerequisite.
3. Add Generate report, Generate dashboard, Generate email and Generate letter steps using their published jobs. Set Refresh source as prerequisite for each. If you want a strictly sequential demo, make each step depend on the preceding step.
4. Use jobs in the same domain with the same approved execution identity. Inspect every version and remove cycles or missing prerequisites.
5. Save draft. Preview the dependency plan and current bindings. Submit for review; Checker performs its own preview and approves/returns; publish the approved batch.
6. Use the published batch’s run/start action with a reason. Open Batch runs → View steps and follow each job run reference and completion checkpoint.
7. Note that ordering the dashboard after a report job does not make the dashboard consume that prior run’s output automatically; dashboard execution invokes its own pinned report definitions.

SCREEN 10

Batch steps and prerequisites

Existing application recorded test · example records may differ from your tenant · click to enlarge

MediFlex > Core > Batches

View A B C D E Overview

← Back Core **Create batch** Save Draft

1 · Batch details

Name the batch and set a bound on its execution. Each job keeps its own retries, permissions and timeout.

Batch code	Batch name	Batch description	Concurrent steps
# BROWSER_1792376991056	Browser batch qbox 17923769	T Browser-authored dependency	# 1

Run deadline seconds

3600

2 · Steps and prerequisites + Add Step

Choose published jobs from this domain with the same execution identity. A step waits for all selected prerequisites to succeed. Leave prerequisites empty to start it immediately.

Step 1 · First report Remove

Step name	Published job	Add prerequisite
First report	Boot acceptance VISTA · Version 1	Choose a step

Starts when the batch is admitted.

Step 2 · Dependent report Remove

Step name	Published job	Add prerequisite
Dependent report	Boot acceptance VISTA · Version 1	Choose a step

Remove Prerequisite: First Report

- A step waits for all selected prerequisites to succeed.
- All selected published jobs must have compatible scope and execution identity.

Expected result

The source refresh completes before the dependent content jobs are admitted. A failure prevents dependent steps from being treated as successful.

Batch dependencies enforce execution order. They do not themselves implement attachment transfer, arbitrary step-output-to-input mapping, record-by-record fan-out or cross-system transaction rollback. If a later step needs those semantics, use a registered operation designed for them.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

Additional demonstration · Step 3 · Pause and resume a batch using checkpoints

Before you act	Required context
Screen / entry	Batches → Batch runs → selected run
Who acts	Authorized batch operator
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

Who: Authorized batch operator

Open: Batches → Batch runs → selected run

1. Open View steps and show the job run reference, attempt generation, status and completion checkpoint for each step.
2. On a prepared isolated run, choose Pause when eligible. Accepted jobs can finish; future admission is paused.
3. Before Resume, use the run's Preview and inspect unresolved failures or external effects. Resume only when the server presents it as eligible.
4. Show that succeeded-step checkpoints are retained and are not blindly repeated. A failed step can be retried only within the job's permitted recovery policy.
5. Verify Stop admission with a reason on an isolated run if required. Note that this does not reverse already completed business operations.
6. Refresh progress and compare the retained failure/recovery history with final step results.

SCREEN 11

Completed Vista, letter and email steps with checkpoints

Existing application recorded test · example records may differ from your tenant · click to enlarge

The screenshot displays the MediFlex Operations Workspace interface. The left sidebar shows a navigation menu with options like 'Your Workspace', 'Find a screen...', 'Business journeys', 'My Dashboard', 'Core', 'Scheduler', 'Job Definitions', 'Schedules', 'Job Runs', 'Business Calendars', 'Queues & Workers', 'Backfills', 'Batches', 'Pulse', 'Vista', and 'Nexora'. The main content area shows the 'Reviewed Vista and Pulse batch e21868 · Run 1' page. It indicates that the run was 'SUCCEEDED · 3 of 3 steps completed.' and provides a table of the steps.

Step Name	Attempt Generation	Status	Job Run Reference	Completion Checkpoint
VISTA acceptance step	1	SUCCEEDED	13223a0d-a880-4708-9f7f-fef7bc9fce32	2998d4d8-fa01-4ce8-ac53-5e59913622c0
LETTER acceptance step	1	SUCCEEDED	17672c0e-ff5f-4276-9881-32a69c574a68	3db0c33f-c4fd-4192-9ec9-d26ce0088609
EMAIL acceptance step	1	SUCCEEDED	17f81de1-cceb-4dc4-b4d7-4bf59e252f97	5cc0204c-4d94-499e-a9c7-b25957d05287

Showing 1-3 of 3

Earlier failed attempts remain in this record. A completion checkpoint records a finished job; business operations retain their own internal progress and compensation rules.

- This is an earlier isolated execution recorded test, not a new Azure run for this guide.
- The checkpoint represents a completed job; the producer owns its internal progress.

Expected result

Recovery preserves successful work and makes unfinished or uncertain work explicit. The final batch outcome is supported by each child job result.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

CHAPTER 29

Additional demonstration · Step 4 · Update, deactivate and retain historical records

Before you act	Required context
Screen / entry	Update, deactivate and retain historical records
Who acts	Authorized operator / reviewer
You need	Complete the previous step in this route, or reopen its eligible saved record. Confirm the same scope and revision.

Steps and field guidance

1. Create a new version of a report/template when business content changes. Save, preview, independently review and publish that version.
2. If a dashboard should use the new report visual, update the dashboard definition through its versioned flow; otherwise it retains its old pinned report version.
3. Create and publish new job versions selecting the new content. Explicitly update/recreate schedules and batch definitions through their supported lifecycle to select the intended versions.
4. Use Deactivate/Reactivate only on records that expose those actions, such as source connections. Use the module's Suspend/Resume/Retire actions where that is the defined lifecycle.
5. Inspect the impact on consumers before disabling a shared source. Current access/availability checks may block later preview or execution.
6. Preserve definitions, approvals, prior versions, outputs and run audit. Pause/retire walkthrough schedules and remove further admissions after the presentation.

Availability, publication, suspension and retirement are different concepts. An `is_active` flag is not a substitute for approval state or an instruction to erase past history. Use the screen's offered, permission-controlled lifecycle action.

Expected result

The demo leaves no recurring job unintentionally enabled and demonstrates explicit version adoption instead of silent changes to approved automation.

Keep the record number and version, any preview/decision/output reference, and the observed result. If the expected result is missing, stop this route and use the recovery chapter; a visible button alone is not completion.

Cross-module handoffs and the record references to retain

Next module	What to carry across	Open the related demonstration
Vista	Follow report/dashboard run references to the actual producer output.	Vista
Pulse	Distinguish generated content from separate dispatch evidence.	Pulse
IAM	Execution ownership and current permissions remain effective at run time.	IAM
Nexora	Inspect campaign renewal health at its learning owner.	Nexora

CHAPTER 31

Problem → responsible action → safe continuation

Procedure	Observed problem	Recovery
Define and preview an operation with an accountable owner	An external operation is absent	Have its owner register and govern the operation contract; Scheduler is not an unrestricted remote-code launcher.
Define and preview an operation with an accountable owner	Preview succeeds but execution later denies access	Recheck execution-time identity, active records and current source authority.
Run once and reconcile the producer result	The start response is lost	Find the existing run before requesting another; remote side effects may already exist.
Run once and reconcile the producer result	Reconciliation required is shown	Inspect producer state and side effects before the permitted recovery action.
Create a calendar-aware one-time or recurring schedule	The next time is surprising	Compare timezone, calendar exclusion and missed-execution rule before changing the job.
Create a calendar-aware one-time or recurring schedule	The calendar changes	Review the resulting schedule impact and adopt the controlled version through the supported path.
Choose retry, rerun, cancel, pause or retire correctly	The producer outcome is uncertain	Reconcile before retry/rerun; a timeout is not proof no external work occurred.
Choose retry, rerun, cancel, pause or retire correctly	An action is disabled	Check current state, policy and permission rather than changing status directly.
Govern queue capacity and inspect worker leases	Jobs remain queued	Check admission state, capacity, worker heartbeat and operation compatibility.
Govern queue capacity and inspect worker leases	A worker disappears during work	Inspect lease expiry/reconciliation and producer outcome before re-admission.

Procedure	Observed problem	Recovery
Preview and admit only missing historical occurrences	All rows say Keep existing	That plan creates no replacements; choose an actual missing training interval if a new-run demonstration is required.
Preview and admit only missing historical occurrences	The preview became stale	Refresh and review the current plan rather than enqueueing outdated authority.
Run dependencies and resume from successful checkpoints	A child has uncertain side effects	Reconcile that producer before resuming dependent work.
Run dependencies and resume from successful checkpoints	A checkpoint is missing	Inspect the actual child run/result; do not manually mark it successful.
Verify the complete automation trail	Producer evidence is missing	Investigate the owning module rather than treating Scheduler completion as sufficient.
Verify the complete automation trail	Verification reports a concern	Preserve the original records and escalate the exact failed boundary; do not rewrite audit rows.

CHAPTER 32

Close the demonstration and hand over the operating guide

Return to the final owning record and show its actual result. Retain the exact references for creation, decision, publication, execution and output that this module produced. Explain any unresolved dependency as a specific blocker; do not convert a source-level example into a claim of a completed hosted journey.

Procedure	Evidence the audience should have seen
Define and preview an operation with an accountable owner	Show registered operation, owner, pinned content and non-executing preview.
Run once and reconcile the producer result	Show admission, run, attempt and actual producer output.
Create a calendar-aware one-time or recurring schedule	Show local/UTC preview, calendar effect, policy and one admitted occurrence.
Choose retry, rerun, cancel, pause or retire correctly	Show the difference between original-run retry, linked rerun and paused future admission.
Govern queue capacity and inspect worker leases	Show queue policy impact, worker heartbeat and a claimed run.
Preview and admit only missing historical occurrences	Show at least one existing occurrence preserved and one actual missing occurrence admitted when prepared.
Run dependencies and resume from successful checkpoints	Show dependency graph, checkpoint, bounded pause/resume and retained successful work.
Verify the complete automation trail	Show definition → timing/admission → run/attempt → producer → recovery/audit lineage.

Keep the training data and audit trail until the agreed review is complete. Use only supported inactivation/retirement actions when authorized; do not erase evidence to reset the demonstration. Hand over the separate user manual for daily operation and record actual results in the UAT pack.