

Salesforce Admin Certification Study Guide

Automation: 15%

■ AUTOMATION OVERVIEW

Why Automation Matters:

- Reduces manual, repetitive work
- Increases user productivity
- Improves platform adoption
- Ensures consistency and accuracy
- Reduces human error
- Frees up time for strategic work

Exam Focus: You'll need to select the RIGHT automation tool based on business requirements and understand the features/limitations of each tool.

Key Topics:

1. Flow Builder (Screen Flows, Record-Triggered Flows, Autolaunched Flows)
2. Approval Processes (Entry criteria, Approvers, Actions)
3. Order of Execution (How flows execute relative to triggers and validation rules)

■ SALESFORCE AUTOMATION TOOLS - EVOLUTION

The Automation Timeline

Past (Being Retired):

- Workflow Rules - Simple automation (deprecated, no longer can create new ones)
- Process Builder - More powerful than workflows (deprecated, being retired)

Present & Future:

- Flow Builder - The ONE tool Salesforce is investing in
- Approval Processes - Still actively

supported and enhanced

Critical Exam Point:

- Workflow Rules and Process Builder are being retired (end of 2025/2026)
- Flow Builder is the future - Salesforce focuses ALL development here
- Learn Flow Builder - it's what you'll use going forward
- Approval Processes are still fully supported

■ FLOW BUILDER

What Is Flow Builder?

Definition: A declarative (point-and-click) automation tool that lets you build complex business logic without code

Key Characteristics:

- Visual interface with drag-and-drop elements
- No coding required (but uses concepts similar to programming)
- Can collect data from users OR run in background
- Can create, read, update, delete records across multiple objects
- Supports complex logic: loops, decisions, formulas
- 10x faster than Process Builder for record updates
- Unified tool replacing Workflow Rules and Process Builder

Where to Access:

- Setup → Flows → New Flow
- Or Setup → Process Automation → Flows

Flow Types - THE MOST IMPORTANT EXAM TOPIC

There are 5 main flow types, but the exam focuses on 3:

1. Screen Flow

Purpose: Requires user interaction through screens

Key Characteristics:

- Has UI elements (screens, forms, wizards)
- Users see and interact with the flow
- Collects data from users
- Displays information to users
- Guides users through multi-step processes
- Cannot be triggered automatically - must be manually launched
- Does NOT support Pause elements

When to Use:

- Guided data entry (wizards)
- Multi-step forms
- Interactive processes requiring user input
- Collecting information before creating records
- Displaying dynamic information to users

How Users Access Screen Flows:

- Lightning pages (embed on record pages, home pages, app pages)
- Custom buttons on page layouts or Lightning pages
- Quick Actions
- Utility bar in Lightning apps
- Lightning Web Components
- Direct URL

Components Available:

- Screen elements (display forms)
- Text, sections, radio buttons, checkboxes, picklists
- File upload
- Display text/images
- Choices and dynamic choices

Real-World Examples:

- New hire onboarding wizard: Step-by-step form collecting employee information
- Customer quote generator: Interactive form calculating quotes based on user selections

- Case escalation form: Guided process for escalating support cases
- Contract renewal wizard: Multi-screen process for renewals

Exam Scenarios:

- "Users need to fill out multi-step form" → Screen Flow
- "Create wizard for data collection" → Screen Flow
- "Guide sales reps through qualification process" → Screen Flow

2. Record-Triggered Flow

Purpose: Runs automatically when a record is created, updated, or deleted

Key Characteristics:

- No user interaction (runs in background)
- Triggered by record changes (DML events)
- 10x faster than Process Builder for updates
- Can run BEFORE or AFTER record is saved
- Most commonly used automation type
- Replaces Process Builder functionality

When to Use:

- Auto-populate fields when record created
- Validate data before saving
- Update related records
- Send notifications
- Create child records
- Enforce business rules

Trigger Types:

A Record Is Created

- Runs when new record is inserted
- Example: Create welcome task when new Contact created

A Record Is Updated

- Runs when existing record is modified

- Can specify which fields trigger the flow
- Example: Update Account when Opportunity closes

A Record Is Created or Updated

- Runs on both insert and update
- Most flexible option
- Example: Validate email format on Contact (create or update)

A Record Is Deleted

- Runs when record is deleted (less common)
- Limited functionality (can't modify the deleted record)

Timing Options:

Before Save (Fast Field Updates):

- Runs BEFORE record is committed to database
- 10x faster than after-save
- Can modify the SAME record (triggering record)
- Cannot:
 - Create/update other records
 - Send emails
 - Use certain actions
 - Use for: Validating data, auto-populating fields on same record

After Save:

- Runs AFTER record is committed to database
- Can perform ALL actions
- Can create/update related records
- Can send emails, post to Chatter
- Use for: Creating related records, notifications, complex logic

Entry Conditions:

- Define which records trigger the flow
- Filter by field values (e.g., Status = "New", Amount > 10000)
- Can use formulas for complex conditions
- Only records meeting criteria run the flow

Real-World Examples:

- Before Save: Auto-calculate discount percentage based on order amount
- After Save - Create: Send welcome email when new customer signs up
- After Save - Update: Create renewal opportunity 90 days before contract expires
- After Save - Create/Update: Sync changes to external system

Exam Scenarios:

- "Auto-update field when record is saved" → Record-Triggered Flow (Before Save)
- "Create task when opportunity closes" → Record-Triggered Flow (After Save - Update)
- "Send email when case is created" → Record-Triggered Flow (After Save - Create)

3. Autolaunched Flow (No Trigger)

Purpose: Runs in background without user interaction, launched by something else

Key Characteristics:

- No trigger defined within the flow itself
- No user interaction (no screens)
- Runs when explicitly invoked by:
 - Another flow (subflow)
 - Process Builder (legacy, being retired)
 - Apex code
 - REST API
 - Custom buttons/links (with parameters)
 - Sits dormant until something starts it
 - Background processing only

When to Use:

- Reusable logic called from multiple places
- Complex calculations invoked by other automations
- API integrations
- Button-triggered actions (assign record to me, mass update)

How to Launch:

1. Subflow: Call from another flow

2. Custom Button/Link: Add button to page layout/Lightning page
3. Apex: Flow.Interview.FlowName
4. REST API: External systems trigger
5. Process Builder (deprecated): Legacy method

Real-World Examples:

- Subflow: Calculate shipping cost (called from multiple parent flows)
- Custom Button: "Assign Case to Me" button on Case page
- Apex: Complex approval routing logic
- API: External system creates order in Salesforce

Exam Scenarios:

- "Reusable logic across multiple processes" → Autolaunched Flow (as subflow)
- "Custom button to perform action" → Autolaunched Flow + Custom Button
- "External API needs to trigger automation" → Autolaunched Flow + REST API

4. Schedule-Triggered Flow (Brief Overview)

Purpose: Runs at specified time and frequency

When to Use:

- Daily/weekly batch processing
- Scheduled reminders
- Data cleanup
- Report generation

Example: Send payment reminders every Monday at 9 AM

5. Platform Event-Triggered Flow (Brief Overview)

Purpose: Runs when platform event is received

When to Use:

- Real-time integrations
- Event-driven architecture

- IoT device notifications

Example: Create case when IoT sensor detects equipment failure

■ ORDER OF EXECUTION - CRITICAL FOR EXAM

What Is Order of Execution?

Definition: The sequence of events Salesforce follows when saving a record

Why It Matters:

- Determines when automations run
- Affects what data is available
- Prevents conflicts between automations
- Critical for troubleshooting
- Exam loves to test this!

The Complete Order of Execution

When you save a record (Insert, Update, Upsert), Salesforce follows this sequence:

1. Load Initial Record

- Loads original record from database (if updating)
- Or loads new record with provided values (if inserting)

2. System Validation

- Checks page layout-specific rules (if from UI)
- Validates field definitions (data types, length)
- Checks required fields

3. Before-Save Record-Triggered Flows ■

- Flows configured to run BEFORE save
- Can modify the triggering record
- Run BEFORE triggers

4. Execute Before Triggers (Apex Code)

- Runs before insert/update/delete triggers
- Custom Apex logic

5. System Validation (Again)

- Required fields must have values
- Field format validation

6. Custom Validation Rules

- Your validation rules execute
- If fail → record is NOT saved

7. Duplicate Rules

- Check for duplicate records
- If block action → stop here

8. Save to Database (NOT COMMITTED YET)

- Record is saved but NOT permanent
- Still in transaction

9. Execute After Triggers (Apex Code)

- Runs after insert/update/delete triggers
- Record already in database but not committed

10. Assignment Rules

- Case assignment rules
- Lead assignment rules

11. Auto-Response Rules

- Sends auto-response emails

12. Workflow Rules (Deprecated)

- Immediate workflow actions
- Field updates re-trigger some steps

13. If Workflow Updated Fields → Repeat Some Steps

- Before/After Update triggers run AGAIN (only once more)
- System validation runs again

- NOTE: Custom validation, flows, duplicate rules do NOT run again

14. Escalation Rules

- Case escalation rules

15. Process Builder (Deprecated)

- Executes Process Builder logic

16. After-Save Record-Triggered Flows ■

- Flows configured to run AFTER save
- Record already committed
- Can create/update other records

17. Entitlement Rules

- SLA and entitlement management

18. Roll-Up Summary Fields

- Calculates summaries on parent/grandparent records
- Parent records go through THEIR save procedure

19. Criteria-Based Sharing Rules

- Evaluates who can see records

20. Commit to Database ■

- ALL changes are now permanent
- Transaction completes

21. Post-Commit Logic

- Sends emails
- Asynchronous Apex (@future, Queueable, Batch)
- Outbound messages

Key Order of Execution Points for Exam

Flow Placement:

- Before-Save Flows: Run in step 3 (before triggers, before validation)
- After-Save Flows: Run in step 16 (after commit, but parent updates can cascade)

What Runs Multiple Times:

- Before/After triggers run TWICE if workflow updates fields
- System validation runs multiple times
- What does NOT run again: Custom validation, flows, duplicate rules, processes, escalation rules

Common Exam Questions:

1. "Validation rule fails even though trigger updated field"
- Answer: Validation rules run BEFORE triggers modify data (step 6 vs step 4)
2. "Flow should run before validation rule"
- Answer: Use Before-Save Flow (step 3) - runs before validation (step 6)
3. "Need to update same record without recursion"
- Answer: Use Before-Save Flow (modifies record before save)
4. "Need to create related record when opportunity closes"
- Answer: Use After-Save Flow (step 16) - can create other records
5. "Workflow updates field, does validation rule run again?"
- Answer: NO - validation rules do NOT run again after workflow field updates

Visual Memory Aid:

Think: "V-BF-BT-V-D-S-AT-A-A-W-[REPEAT]-E-P-AF-E-R-C-COMMIT-POST"

- V = Validation
- BF = Before Flows
- BT = Before Triggers
- D = Duplicate Rules
- S = Save
- AT = After Triggers
- A = Assignment/Auto-response
- W = Workflows
- AF = After Flows
- R = Rollups

■ FLOW BUILDER ELEMENTS

Core Elements You'll Use

Interaction Elements:

- Screen: Display form to users (Screen Flows only)
- Pause: Pause flow until criteria met or time passes

Logic Elements:

- Decision: Branch based on conditions (if/then logic)
- Loop: Iterate through collection of records
- Assignment: Set variable values

Data Elements:

- Get Records: Query records from Salesforce
- Create Records: Insert new records
- Update Records: Modify existing records
- Delete Records: Remove records

Action Elements:

- Send Email: Email notifications
- Post to Chatter: Social collaboration
- Submit for Approval: Start approval process
- Quick Action: Invoke quick actions
- Apex: Call Apex class methods

■ APPROVAL PROCESSES

What Are Approval Processes?

Definition: Automated workflow for getting records reviewed and approved/rejected by designated approvers

Purpose:

- Standardize approval workflows
- Ensure proper sign-off
- Maintain audit trail

- Reduce manual follow-up
- Enforce compliance

Common Use Cases:

- Discount approvals (> 20% discount needs manager approval)
- Expense report approvals
- Contract approvals
- PTO/leave requests
- Purchase orders
- Price exceptions
- Campaign budget approvals

Key Components of Approval Processes

1. Entry Criteria

Purpose: Define which records qualify to enter the approval process

How to Define:

- Formula-based: Write formula (like validation rules)
- Filter-based: Field conditions (Status = "Pending", Amount > 10000)
- All records: No filtering

Examples:

- Opportunities with Amount > \$50,000
- Expense reports where Type = "Travel"
- Discounts greater than 15%
- Contracts with Annual Value > \$100,000
- Cases where Priority = "High"

Best Practices:

- Keep criteria simple and clear
- Test thoroughly before activating
- Document the business logic

- Only include records that NEED approval (reduces workload)

Exam Scenarios:

- "Only opportunities over \$50K need approval" → Entry Criteria: Amount > 50000
- "All expense reports need approval" → Entry Criteria: All records

2. Initial Submission Actions

What Happens When Record Is Submitted:

- Lock Record: Prevent editing while in approval (default)
- Field Updates: Auto-populate fields (e.g., Status = "Pending Approval")
- Email Alerts: Notify submitter
- Tasks: Create task for tracking
- Outbound Messages: Notify external systems

Default Behavior:

- Record is LOCKED when submitted (only submitter/approver can edit)
- Record Owner can't edit unless you unlock

3. Approval Steps

What Are Approval Steps:

- Stages in the approval process
- Define WHO approves at each stage
- Can have multiple sequential steps
- Each step can have its own criteria

Step Criteria:

- All records enter this step: No filtering
- Records meet certain criteria: Filter which records go to this step
- Else action: What happens if criteria not met (go to next step or reject)

Approval Step Order:

- Think of it as a funnel

- Broadest approvals first (most records)
- Narrowest approvals last (fewest records)

Example: Free Trial Approval

- Step 1: VP of Sales approves ALL free trials (broadest)
- Step 2: CFO approves if trial value > \$50K (filtered)
- Step 3: CEO approves if trial value > \$250K (narrowest)

Note: Final step has NO "else" option - if record doesn't meet criteria, it's automatically rejected

4. Approvers - WHO Approves?

Assigning Approvers:

A. Specific User or Queue:

- Assign to a named user (e.g., "John Smith")
- Assign to a queue (e.g., "Management Queue")
- Simplest option but least flexible

B. Related User:

- Assign based on record relationship

- Examples:

- Opportunity Owner's Manager

- Account Owner

- Case Contact's Account Owner

- Record Creator's Manager

- Most common and flexible

C. User Field on Record:

- Record has a lookup field to User (e.g., "Approver" field)
- Submitter selects approver when submitting
- Dynamic and flexible

D. Automatically Based on Record Attributes:

- Queue based on field values
- Example: Route to "West Region Queue" if Account State = CA, OR, WA

- Complex logic for routing

Approval Assignment Options:

Automatically assign using:

- Let submitter choose (user selects from list)
- User field on record (use value from lookup field)
- Manager of record creator
- Manager of record owner
- Queue
- Related user (field on record)

Multiple Approvers:

- Require unanimous approval: ALL approvers must approve
- First response approves/rejects: First person's decision counts
- Approve or reject based on majority: > 50% must approve

Exam Scenarios:

- "Manager must approve their team's expenses" → Related User: Record Creator's Manager
- "Department head approves based on account region" → Automatically assign to Queue
- "User selects approver when submitting" → Let submitter choose

5. Approval Actions

Four Types of Actions:

A. Initial Submission Actions

- Happen when record is FIRST submitted
- Default: Lock record
- Example: Update Status to "Pending Approval", send email

B. Approval Actions (Step-Level)

- Happen when record is APPROVED at this specific step
- If multi-step: Record moves to next step
- If final step: Record goes to final approval actions
- Example: Update field, send notification

C. Rejection Actions

- Happen when record is REJECTED at any step
- Typically unlocks record, updates status, notifies submitter
- Example: Status = "Rejected", unlock record, email submitter

D. Recall Actions

- Happen when submitter RECALLS the approval request
- Always unlocks record (automatic)
- Example: Status = "Recalled", send notification

E. Final Approval Actions

- Happen when record completes ALL approval steps
- Update final status, unlock (if needed), notify stakeholders
- Example: Status = "Approved", unlock record, create contract

E. Final Rejection Actions

- Happen when record is finally rejected
- Clean up, notify, update status
- Example: Status = "Denied", close opportunity

Available Actions:

- Field Updates: Change field values
- Email Alerts: Notify users
- Tasks: Create follow-up tasks
- Outbound Messages: Notify external systems
- Launch Flow: Trigger additional automation (NEW!)

6. Record Locking

Default Behavior:

- Record is LOCKED when submitted for approval
- Only approver and system admin can edit
- Prevents changes while under review

Who Can Edit Locked Records:

- System Admin (Modify All permission)

- Record Approver
- Users in approval delegation chain

Unlocking Options:

- Automatically unlock when approved
- Automatically unlock when rejected
- Keep locked after approval (rare)

Exam Tip: Record locking ensures data integrity during approval process

■ EXAM STRATEGY & KEY CONCEPTS

Critical Distinctions

1. Screen Flow vs. Record-Triggered Flow
 - Screen Flow: Requires user interaction, has screens, manually launched
 - Record-Triggered Flow: No interaction, automatic, triggered by record changes
2. Before-Save vs. After-Save Flows
 - Before-Save: Modifies same record, 10x faster, limited actions, runs before validation
 - After-Save: Can create other records, all actions available, runs after commit
3. Autolaunched Flow vs. Record-Triggered Flow
 - Autolaunched: No trigger, manually invoked by external source
 - Record-Triggered: Has trigger, automatic when record changes
4. Flow Builder vs. Approval Processes
 - Flow: General automation, flexible, any logic
 - Approval Process: Specific to approvals, standardized workflow
5. Approval Entry Criteria vs. Step Criteria
 - Entry Criteria: Determines if record enters approval process
 - Step Criteria: Determines if record goes to this specific step

Common Exam Scenarios

Scenario 1: "Users need to fill out multi-step form to create account with related contacts"

- Answer: Screen Flow (user interaction required)

Scenario 2: "Auto-populate field when opportunity is created"

- Answer: Record-Triggered Flow (Before Save) - fastest option

Scenario 3: "Send email to account team when opportunity closes"

- Answer: Record-Triggered Flow (After Save - Update)

Scenario 4: "Manager must approve discounts over 20%"

- Answer: Approval Process with entry criteria: Discount__c > 20

Scenario 5: "Button on case to assign case to current user"

- Answer: Autolaunched Flow + Custom Button

Scenario 6: "Create task and update related account when opportunity is won"

- Answer: Record-Triggered Flow (After Save) - can update other objects

Scenario 7: "Validation rule fails but flow already updated the field"

- Answer: Order issue - validation runs after before-save flows, use before-save flow

Scenario 8: "Calculate commission before saving opportunity"

- Answer: Record-Triggered Flow (Before Save) - modifies same record

Scenario 9: "Regional managers approve based on opportunity region"

- Answer: Approval Process with dynamic approver assignment (queue or related user)

Scenario 10: "Reusable discount calculation used across multiple flows"

- Answer: Autolaunched Flow (called as subflow)

When to Use Which Automation Tool

Use Screen Flow When:

- ■ Need user input
- ■ Multi-step wizard or guided process
- ■ Interactive data collection
- ■ Display dynamic information

Use Record-Triggered Flow (Before Save) When:

- ■ Update same record fields
- ■ Validate data before saving
- ■ Auto-calculate values
- ■ Need fastest performance

Use Record-Triggered Flow (After Save) When:

- ■ Create related records
- ■ Send notifications
- ■ Update other objects
- ■ Post to Chatter
- ■ Submit for approval

Use Autolaunched Flow When:

- ■ Reusable logic (subflows)
- ■ Button-triggered actions
- ■ API-invoked processes
- ■ Called from Apex

Use Approval Process When:

- ■ Need formal approvals
- ■ Multi-step sign-off required
- ■ Audit trail needed
- ■ Record locking during review
- ■ Standardized approval workflow

Quick Reference: Order of Execution

When Does Flow Run?

- Before-Save Flow: Step 3 (before triggers, before validation)
- After-Save Flow: Step 16 (after everything committed)

What Runs Before Flows?

- System validation (page layout rules, field types)

What Runs After Before-Save Flows?

- Before triggers
- Custom validation rules
- Duplicate rules

What Runs Before After-Save Flows?

- Everything up to commit (triggers, validation, assignment, workflows, escalation, processes)

What Runs After After-Save Flows?

- Entitlement rules
- Roll-up summaries
- Sharing rules
- Commit
- Post-commit logic (emails, async Apex)

Flow Builder Best Practices

- Use Before-Save for same-record updates (10x faster)
- Use After-Save for creating related records
- Add clear descriptions to flows
- Test in sandbox before deploying
- Use decision elements for branching logic
- Bulkify your flows (use collections, not loops per record)
- Handle errors with fault paths
- Keep flows simple and focused
- Document complex logic
- Use subflows for reusable logic

Approval Process Best Practices

- Keep entry criteria clear and documented
- Order steps logically (funnel approach)
- Test all paths (approval, rejection, recall)
- Use queues for scalability (not individual users)

- Provide clear notifications at each step
- Document the process for users
- Consider escalation rules for stalled approvals
- Use dynamic approvers (related users) when possible
- Lock records during approval (default behavior)

■ Summary Checklist

Before the exam, make sure you can answer:

Flow Builder

- [] What's the difference between Screen Flow and Record-Triggered Flow?
- [] When do you use Before-Save vs. After-Save flows?
- [] Can Before-Save flows send emails? (No - limited actions)
- [] Can Before-Save flows create related records? (No)
- [] What's the purpose of Autolaunched Flow?
- [] How do users access Screen Flows? (Lightning pages, buttons, actions)
- [] Which flow type is 10x faster for updates? (Before-Save Record-Triggered)
- [] Can flows run when a record is deleted? (Yes, Record-Triggered)

Order of Execution

- [] When do before-save flows run? (Step 3 - before triggers)
- [] When do after-save flows run? (Step 16 - after commit)
- [] Do validation rules run before or after flows? (After before-save flows, step 6)
- [] Do workflows re-run validation rules? (No)
- [] Do workflows re-run flows? (No)
- [] When are records committed to database? (Step 20)

Approval Processes

- [] What is entry criteria? (Defines which records enter approval)
- [] Can you have multiple approval steps? (Yes)
- [] What happens to record when submitted? (Locked by default)
- [] Who can edit locked records? (Approver, System Admin)

- [] What are the 4 action types? (Initial Submission, Approval, Rejection, Recall, Final Approval, Final Rejection)
- [] How do you assign dynamic approvers? (Related User, User Field, Queue)
- [] Can approval step criteria vary? (Yes, each step can have own criteria)
- [] What does "unanimous approval" mean? (ALL approvers must approve)

Tool Selection

- [] When do you use Screen Flow vs. Record-Triggered Flow?
 - [] When do you use Approval Process vs. Flow?
 - [] When do you use Before-Save vs. After-Save?
 - [] When do you use Autolaunched Flow?
-

■ Final Exam Tips

■ Understand the "why" behind automation:

- Flows run automatically = efficiency
- Screen Flows guide users = user experience
- Approval Processes = compliance & audit trail

■ Know the limitations:

- Before-Save flows can't create other records
- Screen Flows can't be triggered automatically
- Autolaunched Flows have no screens

■ Memorize key differences:

- Before vs. After Save (speed, actions available)
- Screen vs. Record-Triggered (user interaction)
- Flow vs. Approval Process (general automation vs. approvals)

■ Order of Execution is tested frequently:

- Before-Save Flows = Step 3
- After-Save Flows = Step 16
- Validation rules = Step 6
- Know what runs multiple times (triggers, validation)
- Know what does NOT run again (flows, duplicate rules)

■ Approval Process scenarios:

- Entry criteria determines who enters
- Step criteria determines routing
- Related User is most flexible approver assignment
- Record locking is default behavior

■ Think like a consultant:

- "What does the business need?"
- "What's the simplest solution?"
- "What provides the best user experience?"
- "What's most maintainable?"

■ Common trap questions:

- "Flow should validate before saving" → Before-Save Flow (runs before validation)
- "Create related record when saved" → After-Save Flow (only type that can create others)
- "User needs to select values" → Screen Flow (only type with user interaction)

Good luck on your exam! ■