Number: AD0-E902
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: AD0-E902

Exam Name: Adobe Workfront Fusion Developer Professional

Exam A

QUESTION 1
To meet compliance standards, a user must include a process that tracks every Workfront project update created by Fusion.
What can they do to address this business requirement?

A. Use reporting on the Last Updated by ID and Last Update Date
B. Update the External Reference ID with User ID and Timestamp
C. Create a Note record related to the record updated

Correct Answer: C
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Problem Summary:
The organization requires a process to track every project update made by Fusion to meet compliance standards.
This process must provide a clear audit trail of updates, including details like user and timestamp.
Option Analysis:
A . Use reporting on the Last Updated by ID and Last Update Date:
While this provides basic reporting, it only reflects the most recent update and does not maintain a comprehensive history of changes over time.
B . Update the External Reference ID with User ID and Timestamp:
Updating the External Reference ID could cause issues if this field is used for other purposes. It is not designed for logging multiple updates.
C . Create a Note record related to the record updated:
Correct. Creating a Note record for each update ensures that every change is logged with relevant details (e.g., user, timestamp, update reason). This approach creates a full audit trail that is easily accessible and reportable.
Why Note Records are Best:
Audit Trail: Notes provide a clear and searchable history of updates for each project.
Compliance: Ensures compliance by documenting who made what changes and when.
Flexibility: Notes can include custom details such as update reasons or additional context, making them more robust than standard fields.
Implementation:
In the Fusion scenario, add a module to create a Note record after each update.
Populate the Note with relevant details, such as:
User ID ({lastUpdatedBy})
Timestamp ({lastUpdateDate})
Description of the change.

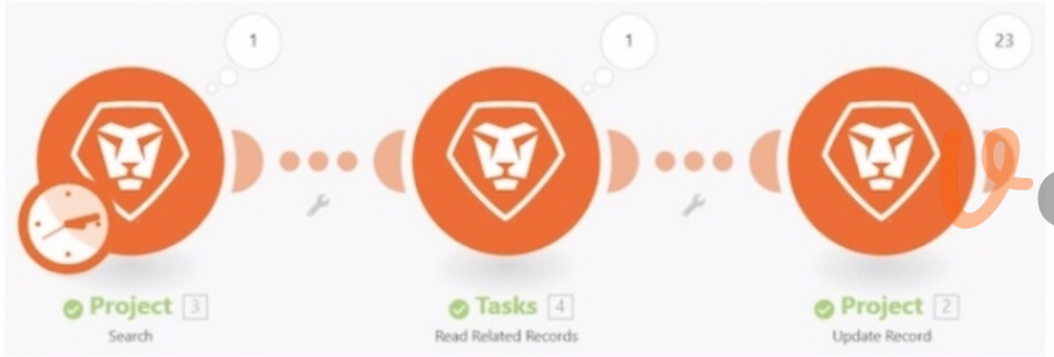
QUESTION 2
Which two actions are best practices for making a Fusion scenario easier to read, share and understand? (Choose two.)

A. Naming all modules by providing short but relevant labels.
B. Insert Note Modules at the beginning of the scenario.
C. Add notes where applicable to clarify what is happening.
D. Attach the requirements document using the scenario settings.

Correct Answer: A, C
Section:
Explanation:

Step by Step Comprehensive Detailed Explanation:
Best Practices for Scenario Clarity:
Workfront Fusion scenarios can become complex. Adopting practices that enhance readability, shareability, and understanding ensures the scenario can be maintained and used effectively by others.
Option Analysis:
A . Naming all modules by providing short but relevant labels:
Correct. Proper naming helps identify the function of each module at a glance. For example, instead of generic names like 'Project Search,' use 'Search High Priority Projects.'
This makes it easier to debug, share, and update the scenario.
B . Insert Note Modules at the beginning of the scenario:
Incorrect. While notes are useful, inserting a Note module at the beginning is not always necessary unless clarification is required for the initial step. Adding notes throughout the scenario (Option C) is more beneficial.
C . Add notes where applicable to clarify what is happening:
Correct. Adding comments or notes helps explain the purpose of certain steps, making the scenario easier to understand for collaborators or when revisiting it in the future.
D . Attach the requirements document using the scenario settings:
Incorrect. Attaching a requirements document might be useful for reference but does not directly contribute to scenario readability or understanding within the interface.
Implementation Tips:
Use descriptive names for modules that clearly indicate their purpose (e.g., 'Update Project Status' instead of 'Update Record').
Add comments or notes at decision points or complex mapping expressions to explain logic.

QUESTION 3
Refer to the exhibit.



This scenario shows a 1 in the bundle inspector for the Tasks module and a 23 in the bundle inspector for the Project module.
What does the number in the bundle inspector represent?

A. The number of seconds to process the module
B. The number of output bundles
C. The number of operations performed
D. The number of times a module has been edited

Correct Answer: B
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Understanding the Scenario:
In Workfront Fusion, each module in a scenario processes data and generates bundles as output.
The bundle inspector shows the number of bundles (data packets) output by a module during an execution.
Option Analysis:
A . The number of seconds to process the module:
This is incorrect. The number in the bundle inspector does not indicate time but rather the count of output bundles. Processing time is not displayed in this way.

B . The number of output bundles:
Correct. The number displayed in the bundle inspector represents how many bundles the module output during the execution. In the given example, the 'Tasks' module outputs 1 bundle, and the 'Project' module outputs 23 bundles.
C . The number of operations performed:
This is incorrect. The bundle inspector displays the number of output bundles, not operations. While operations may be a result of processing bundles, they are tracked separately in Fusion reports.
D . The number of times a module has been edited:
This is incorrect. Editing history is not displayed in the bundle inspector.
Explanation of Bundle Inspector:
Each module processes input data and generates output bundles.
These numbers in the bundle inspector indicate how many bundles the module is outputting in the current run of the scenario.
For example, if a 'Search' module retrieves 23 records, the bundle inspector will show 23, meaning the module outputs 23 bundles.
Context of the Given Image:
The 'Tasks' module processes and outputs 1 bundle.
The 'Project' module processes 1 input bundle (from 'Tasks') and outputs 23 bundles.
Reference: This information is consistent with Workfront Fusion documentation, which explains the bundle inspector's function during scenario execution. The bundle inspector is used to monitor data processing and ensure expected outputs from modules.

QUESTION 4
According to Workfront's training on scenario testing, what are three of the essential elements of a test plan? (Choose three.)

A. Roadmap requirements
B. Description of expected behavior
C. Specific event/trigger per scenario
D. Description of testing steps
E. Executive sponsor expectations

Correct Answer: B, C, D
Section:
Explanation:
Workfront's training on scenario testing emphasizes the importance of a well-structured test plan to ensure scenario reliability and accuracy. The three essential elements include:
B . Description of Expected Behavior:
This provides clarity on what the scenario is supposed to achieve when executed successfully.
It serves as a benchmark for evaluating the outcome of test executions.
C . Specific Event/Trigger per Scenario:
Identifying and testing specific triggers ensures that the scenario starts under the correct conditions.
This is crucial for verifying the proper configuration of the scenario's start point.
D . Description of Testing Steps:
Outlining step-by-step instructions for the testing process ensures that all aspects of the scenario are tested systematically.
It helps identify potential bottlenecks or areas for improvement in the scenario's configuration.
Why Not Other Options?
A . Roadmap requirements: This pertains to project planning and is not directly relevant to scenario testing.
Workfront Training Materials: Best Practices for Scenario Testing
Experience League Documentation: How to Design and Execute a Test Plan for Workfront Fusion Scenarios
E. Executive sponsor expectations: While valuable for overall project alignment, it is not an essential component of a technical test plan.

QUESTION 5
A customer wants all their Salesforce Opportunities to sync with their connected projects in Workfront -approximately 20,000+ projects.
After the admin sets a Workfront Fusion scenario to run each night and perform this action, the scenario is run once to test. After 40 minutes, it unexpectedly stops running.
Why did this occur?

A. Workfront has a limit to the number of API calls it can receive and stopped the scenario from running
B. Workfront Fusion occasionally times out if trying to process over 2000 records within a 40-minute period
C. Workfront Fusion has an execution timeout and likely stopped the scenario from running
D. The Workfront API stops integration webhooks if they are hit more than 2000 times in a 10 minute period

Correct Answer: C
Section:
Explanation:
Understanding the Issue:
The customer is syncing 20,000+ Salesforce Opportunities with Workfront projects using a scheduled Fusion scenario.
After running for 40 minutes, the scenario unexpectedly stops.
Why Option C is Correct:
Workfront Fusion Execution Timeout:
Fusion scenarios have a default execution timeout of 40 minutes per run.
If the scenario exceeds this time limit, Fusion automatically stops the execution to avoid resource overuse.
Handling Large Data Sets:
Scenarios involving large datasets (like syncing 20,000+ records) may require optimizations, such as breaking the data into smaller chunks using paginated requests or iterators.
In this case, the scenario stopped because the execution timeout was reached, not due to API limits or webhook restrictions.
Why the Other Options are Incorrect:
Option A ('Workfront API call limit'):
While Workfront does have API rate limits, they are generally generous and not the reason for the scenario stopping. Fusion scenarios are designed to manage API calls efficiently.
Option B ('Fusion times out if processing over 2000 records in 40 minutes'):
This is incorrect because Fusion does not have a hard limit on the number of records processed in 40 minutes. The timeout is time-based, not record-based.
Option D ('Workfront API stops webhooks after 2000 hits in 10 minutes'):
This does not apply to Fusion scenarios. Webhooks are separate from the API calls initiated by Fusion.
How to Resolve the Issue:
Split the Data: Use pagination or batch processing to divide the 20,000+ records into smaller chunks (e.g., 1,000 or 2,000 records per run).
Adjust Scheduling: Schedule the scenario to run more frequently with smaller batches, ensuring all records are synced over multiple runs.
Use Iterators: Add an Iterator module to loop through smaller subsets of data, preventing the scenario from exceeding the execution timeout.
Steps to Optimize the Scenario:
Add a Search Module to retrieve opportunities in smaller batches (e.g., using limits or pagination parameters).
Use a Repeater Module to process each batch iteratively.
Save the scenario and schedule it to run nightly or more frequently, depending on the sync requirements.
Reference and Supporting Documentation:
Adobe Workfront Fusion: Execution Timeout Limits
Workfront Community: Managing Large Data Sets in Fusion Scenarios
By optimizing the scenario to handle smaller batches of data, the admin can avoid the execution timeout issue and ensure successful syncing of Salesforce Opportunities with Workfront projects.

QUESTION 6
A web service provides the following array named 'Colors':

```
[
   {
      "ID":"22342",
      "name":"Red"
   },
   {
      "ID":"33495",
      "name":"Blue"
   }
]
```

Which expression returns the first ID in the array?

A.

```
map( 2. Colors ; ID ; ID ; 1 )
```

B.

```
get( map( 2. Colors ; ID ) ; 1 )
```

C.

```
map( get( 2. Colors ; ID ) ; 1 )
```

Correct Answer: B
Section:
Explanation:
Understanding the Array and the Task:
Input Array (Colors):
[
{ 'ID': '22342', 'name': 'Red' },
{ 'ID': '33495', 'name': 'Blue' }
]
Goal: Extract the first ID from the array, which is '22342'.
Why Option B is Correct:
The expression get(map(2.Colors; ID); 1):
map(2.Colors; ID): Iterates over the array 2.Colors and extracts the ID field from each object. This creates a new array containing just the IDs: ['22342', '33495'].
get(...; 1): Retrieves the first element of the newly created array, which is '22342'.
Why the Other Options are Incorrect:
Option A (map(2.Colors; ID; ID; 1)):
This syntax is invalid because the additional ID and 1 parameters are misplaced. The map function requires only two arguments: the array and the field to map.
Option C (map(get(2.Colors; ID); 1)):
This incorrectly attempts to use get inside map. The get function does not return a field for mapping, so the syntax is invalid.
How the Expression Works:
Step 1: map(2.Colors; ID)
Extracts the ID field from each object in the Colors array.
Output: ['22342', '33495'].

Step 2: get(...; 1)
Retrieves the first element of the mapped array.
Output: '22342'.
Use Case in Workfront Fusion:
This approach is commonly used when processing arrays in Fusion scenarios, ensuring specific elements are accessed without additional looping or complex logic.
Reference and Supporting Documentation:
Adobe Workfront Fusion Functions Documentation
Workfront Community: Using Map and Get Functions
By combining map and get, this expression efficiently extracts the first ID from the array, ensuring correct and reliable results.

QUESTION 7
Refer to the exhibit.
In this image,

all items are displayed in a position that does not hide their execution sequence. In what order do the modules execute?

A. Google Drive > Email > Workfront > Data Store > Salesforce
B. Email > Workfronl > Data Store > Salesforce > Google Drive
C. Email > Data Store > Workfront > Salesforce >
D. Google Drive Google Drive > Email > Data Store > Workfront > Salesforce

Correct Answer: D
Section:
Explanation:
Understanding the Diagram:
The image depicts a Workfront Fusion scenario using Routers to split and manage multiple execution paths.
Fusion executes modules in a left-to-right and top-to-bottom sequence along each path. Routers direct the flow to downstream modules.
Determining the Execution Sequence:
Path 1:
The first path begins with Google Drive. The Router branches out, directing execution sequentially to other modules.
Path 2:

The second path starts with Email and continues downwards through the remaining modules.
Order Within Branches:
After passing through a Router, each branch completes its sequence of modules before moving to the next Router.
Why Option D is Correct:
Execution begins from the leftmost module (Google Drive) and flows rightward. The sequence is as follows:
Google Drive
Email
Data Store
Workfront
Salesforce
This matches the visual layout and Fusion's execution rules for scenarios with Routers.
Why the Other Options are Incorrect:
Option A ('Google Drive > Email > Workfront > Data Store > Salesforce'):
Incorrect because Workfront does not precede Data Store in the execution order.
Option B ('Email > Workfront > Data Store > Salesforce > Google Drive'):
Incorrect because execution begins with Google Drive, not Email.
Option C ('Email > Data Store > Workfront > Salesforce'):
Incorrect because it excludes Google Drive as the starting point and places modules out of sequence.
Execution Rules in Workfront Fusion:
Fusion executes modules in the order they appear visually, starting from left to right and top to bottom within each branch.
Routers split the execution into separate branches, which are completed sequentially.
Reference and Supporting Documentation:
Adobe Workfront Fusion: Execution Flow Rules
Workfront Community: Understanding Router and Module Execution
The correct execution sequence is Google Drive > Email > Data Store > Workfront > Salesforce, ensuring all paths are processed as per the scenario's layout.

QUESTION 8
A Fusion scenario updates project conditions each night, and should set the project condition to At Risk if there are any high priority open issues on the project. The scenario retrieves all open projects and cycles through the projects. For each project with issues, it retrieves all associated open issues, iterates through them and sets the project condition to At Risk if the issue is high priority or On Target if it is not.
A user notices that Fusion is updating the progress condition multiple times, once for each issue in the project.
How can the developer ensure the project is updated only once?

A. Change the issue search module to result set of First Matching
B. Record Add an Ignore error directive as an error handler route for the update module
C. Create a separate scenario to update the overall project condition
D. Apply the Run Once flow control function

Correct Answer: C
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Problem Summary:
The Fusion scenario updates the project condition multiple times, once for each high-priority issue.
The desired behavior is to update the project condition only once, based on the overall condition of all associated issues.
Option Analysis:
A . Change the issue search module to result set of First Matching:
This would limit the search to only the first issue. However, this does not account for all issues on the project, leading to incomplete logic for setting the project condition.
B . Add an Ignore error directive as an error handler route for the update module:
Ignoring errors does not prevent multiple updates; it only suppresses errors in the workflow.

C . Create a separate scenario to update the overall project condition:
Correct. By separating the project update logic into a different scenario, the developer can ensure the condition is updated only once after analyzing all issues. The project condition is calculated holistically, based on the state of all high-priority issues.
D . Apply the Run Once flow control function:
'Run Once' controls execution at the scenario level, not within a module's iteration. It cannot prevent multiple updates in this context.
Why Separate Scenario is Best:
Simplifies Logic: A separate scenario can be designed to run after all issues have been checked, ensuring only one update per project.
Avoids Redundancy: Prevents unnecessary API calls to update the same project multiple times.
Improves Performance: Reduces the number of operations and bundles processed in the main scenario.
Implementation:
Create a separate scenario triggered after the issue-checking scenario completes.
Use aggregate data (e.g., a data store or intermediate processing) to evaluate the overall project condition before performing a single update.

QUESTION 9
A Fusion designer needs to create a Fusion scenario that will assign a user to each task with the Copywriter job role on a project.
Which method results in the fewest number of operations?

A. Searching for all assignments in the project where the role is Copywriter and add the user to each assignment
B. Searching for Tasks in the project with the Copywriter role and assign the user to each task
C. Using the Misc Action module for the project to assign all Copywriter assignments to the user

Correct Answer: C
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Scenario Context:
The goal is to assign a user to all tasks on a project where the role is Copywriter.
The chosen method should minimize the number of operations to ensure efficiency.
Option Analysis:
A . Searching for all assignments in the project where the role is Copywriter and add the user to each assignment:
This approach involves searching assignments, iterating through them, and adding the user individually. Each iteration generates multiple operations, making it less efficient.
B . Searching for Tasks in the project with the Copywriter role and assign the user to each task:
Similar to Option A, this approach also requires iterating through tasks and assigning users one by one, resulting in a higher number of operations.
C. Using the Misc Action module for the project to assign all Copywriter assignments to the user:
Correct. The Misc Action module can perform bulk actions (e.g., assigning users to roles) in a single operation. This method is the most efficient, as it minimizes the number of operations while achieving the same result. Why the Misc Action Module is Best: Efficiency: Bulk operations reduce the number of API calls and iterations. Performance: Using fewer operations optimizes scenario execution and reduces resource consumption. Simplicity: Avoids the complexity of iterating through tasks or assignments individually. Implementation: Add a Misc Action module to the scenario. Configure the module to assign the user to all tasks with the Copywriter role on the selected project. Test the module to ensure it performs the bulk assignment as expected.

QUESTION 10
A Fusion scenario is making too many requests to a third-party API, which returns a 429 'Too Many Requests' error Which technique reduces the number of API requests?

A. Using a Search module to get record IDs and then read those IDs with a Read Record module to pull other data
B. Moving Search and GET modules earlier in the scenario instead of pulling more data about the same record multiple times
C. Adding a Retry error handling directive to the Fusion scenario

Correct Answer: B
Section:
Explanation:

Understanding the Issue:
The scenario is making too many API requests, causing the third-party API to return a 429 'Too Many Requests' error, which indicates that the rate limit has been exceeded.
The solution needs to reduce unnecessary or redundant API requests to prevent hitting the API limits.
Why Option B is Correct:
Avoid Redundant Requests:
Placing Search and GET modules earlier in the scenario ensures that all required data is retrieved in one batch or in fewer requests, rather than repeatedly querying the same record later in the scenario.
This technique reduces the overall number of API requests sent to the third-party system.
Efficient Data Flow:
By structuring the scenario to retrieve all necessary data at the beginning, subsequent modules can reuse the retrieved data instead of making additional API calls.
Why the Other Options are Incorrect:
Option A ('Using a Search module and then a Read Record module'):
This approach can increase API requests, as the Search module retrieves record IDs, and the Read Record module makes separate API requests for each record. This often results in more requests than necessary.
Option C ('Adding a Retry error handling directive'):
Adding a Retry directive does not reduce the number of requests. Instead, it retries failed requests, which could worsen the problem by increasing API traffic.
Best Practices to Reduce API Requests:
Consolidate data retrieval into a single module or a smaller number of requests.
Use caching or intermediate storage (like Fusion Data Stores) to avoid re-fetching the same data.
Limit the scope of Search modules by using filters or pagination to process smaller, relevant data sets.
Reference and Supporting Documentation:
Adobe Workfront Fusion Best Practices: Managing API Rate Limits
Workfront Community: Error 429 Solutions

QUESTION 11
Which statement about the differences between instant and polling triggers is true?

A. To keep track of records processed, instant triggers store received webhooks in a queue, whereas polling triggers remember which records have already been processed
B. A user should use instant triggers when available because instant triggers allow Fusion to process bundles of data faster than polling triggers
C. A user must set up a webhook in Fusion to use Instant Triggers that make polling triggers easier to use and more reliable in scenarios
D. Only polling triggers can be set to run on a schedule and should be used to avoid shutdown of third-party systems during working hours

Correct Answer: B
Section:
Explanation:
Understanding Instant and Polling Triggers:
Instant Triggers:
Rely on webhooks to receive real-time data from a third-party system.
The external system sends a notification (webhook) to Fusion whenever an event occurs, triggering the scenario immediately.
Polling Triggers:
Regularly check (poll) the third-party system for new or updated records at scheduled intervals.
These are slower because they involve repeated API requests.
Why Option B is Correct:
Speed and Efficiency:
Instant triggers process data faster because they act immediately upon receiving a webhook. Polling triggers, on the other hand, may take time depending on the polling frequency and can result in unnecessary delays.
Reduced Load on Systems:
Instant triggers generate fewer API calls than polling triggers, which continuously check for new records even if no changes have occurred.
Best Practice: Use instant triggers whenever supported by the third-party system to ensure faster and more efficient scenario execution.
Why the Other Options are Incorrect:
Option A ('Instant triggers store received webhooks in a queue'):
Webhooks do not store data in a queue; they simply notify Fusion of events in real-time. Polling triggers also do not store records but remember the last processed record.

Option C ('A user must set up a webhook in Fusion'):
Instant triggers require setting up webhooks in the external system, not in Fusion. Fusion provides the webhook endpoint, but the user must configure the source system to send data.
Option D ('Only polling triggers can be set to run on a schedule'):
This is incorrect because instant triggers do not rely on schedules; they operate in real-time. Polling triggers, however, run on schedules and are used when instant triggers are unavailable.
Reference and Supporting Documentation:
Adobe Workfront Fusion Triggers Documentation
Workfront Community: Differences Between Instant and Polling Triggers
Instant triggers are the preferred option when available, as they provide real-time data processing with greater speed and efficiency than polling triggers.

QUESTION 12
A Fusion designer is unhappy with the high number of bundles passing through an instant Watch Events module that monitors Workfront project updates.
Which action reduces the number of bundles passing through the module?

A. Reducing the maximum number of returned events on the trigger
B. Reducing the maximum number of cycles in scenario setting
C. Changing the module type to Watch Record and applying criteria in the optional filter box

Correct Answer: C
Section:
Explanation:
Understanding the Issue:
The Watch Events module is generating a high number of bundles because it monitors a broad range of project updates in Workfront, resulting in an overwhelming amount of data passing through the scenario.
The goal is to reduce the number of bundles by narrowing the scope of monitored events.
Why Option C is Correct:
Switching to Watch Record:
The Watch Record module allows users to monitor specific records (e.g., projects, tasks) with additional filtering options in the criteria or filter box.
By applying filters, the module can focus only on relevant updates, significantly reducing the number of bundles being processed.
Example: Filtering for specific project statuses, update types, or assigned users ensures that only relevant changes are captured.
Why the Other Options are Incorrect:
Option A ('Reducing the maximum number of returned events on the trigger'):
This limits the number of bundles processed per cycle but does not address the root cause, which is the broad monitoring scope of the Watch Events module.
Option B ('Reducing the maximum number of cycles in scenario settings'):
The number of cycles determines how many iterations the scenario performs in one run but does not reduce the number of bundles entering the scenario.
Steps to Use the Watch Record Module:
Replace the Watch Events module with Watch Record.
Specify the record type to monitor (e.g., Project).
Use the optional filter box to apply criteria, such as specific project fields, statuses, or other conditions.
Activate the scenario to test the refined data flow.
Reference and Supporting Documentation:
Adobe Workfront Fusion: Watch Record Module
Workfront Community: Managing High Bundle Volumes in Fusion

QUESTION 13
Which module must a user select to upload a document into Workfront and attach it to a task?

A. Create Record for Document Version after Create Record for the document on the task
B. Upload Document while setting the related record
C. Create Record of Document type while setting the related record
D. Miscellaneous Action to attach document to a task

Correct Answer: B
Section:
Explanation:
Understanding the Requirement:
The user wants to upload a document into Workfront and attach it to a specific task.
This action involves creating a document in Workfront and associating it with a task as a related record.
Why Option B is Correct:
The Upload Document module is specifically designed for uploading files into Workfront.
It includes the ability to set a related record (e.g., a task, project, or issue) to which the document will be attached.
This ensures the document is uploaded and correctly linked to the task in a single operation.
Why the Other Options are Incorrect:
Option A ('Create Record for Document Version after Create Record for the document on the task'):
This involves multiple steps, which are unnecessary. The Upload Document module already handles both the upload and the attachment in one action.
Option C ('Create Record of Document type while setting the related record'):
The Create Record module is not designed for file uploads. It only creates metadata records, not the actual document.
Option D ('Miscellaneous Action to attach document to a task'):
There is no Miscellaneous Action specifically for attaching a document to a task. The Upload Document module is the appropriate choice.
Steps to Upload a Document in Workfront Fusion:
Add the Upload Document module to the scenario.
Specify the file to upload (e.g., from a previous module like Google Drive or an HTTP request).
Set the related record to the target task by providing its ID.
Run the scenario to upload and attach the document to the task.
Reference and Supporting Documentation:
Adobe Workfront Fusion: Upload Document Module
Workfront Community: Best Practices for Document Management in Fusion
The Upload Document module is the most efficient and accurate method for uploading and attaching a document to a task in Workfront.

QUESTION 14
A source system provides a list of users and their job roles and departments. The destination system requires the job role values be revised to match the job roles by different department. The end users currently use a spreadsheet to manage that mapping. Updates to this list are infrequent, and management requires more security than the current process offers.
Which method is appropriate for this use case?

A. Switch
B. Spreadsheet
C. Data store
D. Table aggregator

Correct Answer: C
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Scenario Summary:
The source system provides a list of users, job roles, and departments.
The destination system requires a mapping of job roles to different departments.
Updates to this mapping are infrequent, and there is a concern for more security than a spreadsheet offers.
Option Analysis:
A . Switch: A switch function in Workfront Fusion is used to make decisions based on specific conditions but does not maintain persistent data like mappings over time. It is not suitable for storing mappings that require infrequent updates.
B . Spreadsheet: Spreadsheets may be simple to use but are less secure and not ideal for integrating with Workfront Fusion workflows due to their lack of direct integration features or control over changes.

C . Data store: A data store in Workfront Fusion is specifically designed for storing persistent data, such as mappings or reference tables. It is secure, easy to manage, and integrates seamlessly with workflows, making it the best choice for this use case.
D . Table aggregator: A table aggregator is used for combining or processing data within a workflow but is not suitable for storing persistent mappings outside of a running scenario.
Why Data Store is Appropriate:
Persistence: A data store is ideal for maintaining the job role and department mappings across scenarios.
Security: It is managed within Workfront Fusion, offering better control and access restrictions compared to a spreadsheet.
Ease of Integration: Data stores can be directly queried or updated in Fusion workflows, allowing seamless handling of mappings.
Scalability: While updates are infrequent, the data store can handle changes easily and scale if the volume of mappings increases.
Implementation in Workfront Fusion:
Create a data store for job roles and departments in Workfront Fusion.
Set up fields for Job Role and Department in the data store.
Use Workfront Fusion scenarios to retrieve and update mappings from this data store based on requirements.
Reference: This approach aligns with the Workfront Fusion documentation on data stores, which are designed for secure and efficient storage of persistent data used across scenarios. See Workfront Fusion resources for data store setup and use cases for further details.

QUESTION 15
A scenario is too large, with too many modules. Which technique can reduce the number of modules?

A. Nesting multiple mapping panel functions instead of setting and resetting variables when transforming data in more than one way
B. Using a Compose a string module to combine variables and module output. Then use the Text Parser to parse the data and assign to variables
C. Setting the scenario to Auto Commit in scenario settings

Correct Answer: A
Section:
Explanation:
Step by Step Comprehensive Detailed Explanation:
Problem Summary:
The scenario has become too large due to the high number of modules.
The goal is to reduce the number of modules by optimizing how data is transformed.
Option Analysis:
A . Nesting multiple mapping panel functions:
Nesting multiple functions in the mapping panel (e.g., using if(), concat(), replace()) eliminates the need for separate modules to set and reset variables for each transformation.
This is a highly efficient technique to transform data in fewer modules, making it the correct answer.
B . Using a Compose a string module and Text Parser:
This involves additional modules (Compose a string + Text Parser) instead of reducing the number of modules. It is not an optimal solution to this problem.
C . Setting the scenario to Auto Commit:
The Auto Commit setting helps with transactional control and does not reduce the number of modules in a scenario.
Why Nesting Mapping Functions is Effective:
Efficiency: Complex transformations can be performed inline within a single mapping panel.
Readability: Proper nesting and naming conventions make it easier to understand the logic without adding unnecessary modules.
Scalability: This approach keeps the scenario compact and reduces complexity as the scenario grows.
How to Implement:
Open the mapping panel in relevant modules.
Use multiple nested functions like if(), concat(), add(), etc., within the mapping expressions.
Test the mapping thoroughly to ensure correctness.

QUESTION 16
A user queried Salesforce for user information, and it returned a name, email address, and user ID. The user would like to assign a task in Workfront.
Which steps are required to assign the task?

A. Assign the task using the provided user ID > add the user ID to the assignment field
B. Query Workfront for user based on email address > assign task using returned ID
C. Query Workfront based on the user's name > assign task using returned email address

Correct Answer: B
Section:
Explanation:
Step 1: Query Workfront for User Based on Email Address
Salesforce provides the email address of the user. Since Workfront uses user IDs for task assignments, the email address can be used as a unique identifier to query Workfront's database.
A query module in Workfront Fusion retrieves the corresponding Workfront user ID using the provided email address.
Step 2: Assign Task Using Returned ID
After retrieving the Workfront user ID, use it in the assignment field of the task module in Fusion.
The task assignment process requires a valid user ID, ensuring proper linkage and assignment within Workfront.
Why Not Other Options?
A . Assign the task using the provided user ID > add the user ID to the assignment field: The Salesforce user ID is not recognized by Workfront. It is necessary to query Workfront to convert the email into a Workfront-compatible user ID.
C . Query Workfront based on the user's name > assign task using returned email address: Workfront uses user IDs, not email addresses, for task assignments. Names are also not unique and could cause assignment errors.
Adobe Workfront Fusion Documentation: Task Assignments Using User IDs
Experience League Community: Mapping Salesforce Data to Workfront Tasks

QUESTION 17
What are two required elements of a test case? (Choose two.)

A. Expected outcome of test
B. Name of test owner
C. Clear procedure for completing the test
D. Source code being tested

Correct Answer: A, C
Section:
Explanation:
A . Expected Outcome of Test
A test case must clearly state what the expected outcome is, providing a standard against which the results can be measured.
This ensures testers can validate whether the scenario behaves as intended.
C . Clear Procedure for Completing the Test
A well-defined procedure outlines the exact steps required to execute the test, ensuring consistent and repeatable testing.
This reduces ambiguity and helps identify whether errors are due to the scenario configuration or improper test execution.
Why Not Other Options?
B . Name of Test Owner: While helpful for accountability, the name of the test owner is not a required component of the test case itself.
D . Source Code Being Tested: Fusion scenarios do not typically involve source code. Instead, the focus is on workflow execution and configuration, making this element irrelevant.
Workfront Training Materials: Test Case Design Best Practices
Adobe Workfront Fusion Documentation: Testing and Debugging Scenarios

QUESTION 18
A source system should provide a Workfront task ID, but queries keep resulting in an error.
Which module helps determine if the task ID is valid?

A. Read Record

B. Read Related Record
C. Search Record

Correct Answer: C
Section:
Explanation:
To determine if a task ID is valid in Workfront, you can use the Search Record module. This module allows you to search for records, such as tasks, within Workfront based on a given ID.
Search Record Module:
This module can be used to search for a specific task in Workfront by its task ID.
If the task ID is valid, it will return the task details; if the task ID is invalid, it will return an error or no results.
The Search Record module is designed to check if a specific record exists in Workfront, making it ideal for verifying the validity of a task ID.
Why Not Other Options?
A . Read Record: This module reads a specific record by ID but does not search across multiple records. It may not provide the flexibility needed to check if a task ID is valid without knowing the exact task ID.
B . Read Related Record: This module is used to read a related record (e.g., reading the project related to a task). It is not suitable for validating a task ID directly.
Adobe Workfront Fusion Documentation: Search Record Module Usage
Experience League Community: Verifying Record IDs in Workfront Fusion

QUESTION 19
A custom API call to a web service is used inside of a high volume iteration. The module that calls the web service sometimes returns an error - 429: Too many requests.
Which two actions may be used to address this error? (Choose two.)

A. Add a sleep module just prior to the module generating the error
B. Use an ignore directive on the module generating the error
C. Add a module to test the service for errors
D. Use a break directive on the module generating the error

Correct Answer: A, C
Section:
Explanation:
When encountering the error 429: Too many requests, which indicates the web service is being overwhelmed by requests, the following actions can help:
A . Add a Sleep Module:
Adding a Sleep module introduces a delay between iterations, reducing the frequency of API calls.
By slowing down the rate of requests, you avoid hitting the rate limits of the web service, thus reducing the chances of receiving a 429 error.
This approach is useful for managing high-volume iterations without overloading the external service.
C . Add a Module to Test the Service for Errors:
Adding a module to test the service's response before making a call can help prevent the 429 error by checking if the service is ready to handle requests.
This preemptive check allows the scenario to conditionally execute, ensuring that it doesn't overwhelm the service and respects the API rate limits.
Why Not Other Options?
B . Use an Ignore Directive: Ignoring errors can be risky because it would cause the scenario to ignore 429 errors, possibly leading to failed API calls that are not addressed. Ignoring an error doesn't solve the issue of too many requests being sent to the service.
D . Use a Break Directive: The Break directive would stop the execution, which is counterproductive when trying to resolve the issue by reducing the rate of requests. It would not address the root cause of too many requests.
Adobe Workfront Fusion Documentation: Handling API Rate Limiting with Sleep and Error Handling
Experience League Community: Managing Web Service Errors in High-Volume Iterations

QUESTION 20
A Fusion user needs to connect Workfront with a third-party system that does not have a dedicated app connector in Fusion.
What should the user do to build this integration?

A. Determine the API structure and authentication protocols for the third-party system and then use the appropriate Universal Connector

B. Create a new connection to the third-party system in the connections area and then the Universal Connectors will be available for use
C. Use the Workfront Custom API module to set up the connection using API calls to the third-party system

Correct Answer: A
Section:
Explanation:
Understanding the Requirement:
If a third-party system does not have a dedicated app connector in Workfront Fusion, users can still build an integration using Universal Connectors.
Universal Connectors in Fusion allow users to configure custom API calls, enabling communication with systems that lack pre-built integrations.
Steps to Build the Integration:
Determine the API Structure: Review the third-party system's API documentation to understand the available endpoints, data formats (e.g., JSON, XML), and request/response structure.
Identify Authentication Protocols: Determine how the third-party system handles authentication (e.g., API keys, OAuth 2.0, Basic Auth).
Configure the Universal Connector: Use modules like HTTP Request or Webhook to make API calls to the third-party system based on the documented structure.
Why Not Other Options?
B . Create a new connection to the third-party system in the connections area and then the Universal Connectors will be available for use: Creating a new connection in the connections area is only applicable for predefined connectors, not for Universal Connectors, which require manual configuration for unsupported systems.
C . Use the Workfront Custom API module to set up the connection using API calls to the third-party system: The Workfront Custom API module is specifically designed for Workfront's own API, not for connecting to third-party systems.
Adobe Workfront Fusion Documentation: Using Universal Connectors for Custom Integrations
Experience League Community: Integrating Third-Party Systems Using Workfront Fusion Universal Modules

QUESTION 21
Which two statements about working with incomplete executions are true? (Choose two.)

A. When incomplete executions are togged, a user must review the History tab to understand the error and then manually run the entire scenario to resolve
B. Incomplete executions are not stored by default. This option must be enabled in the scenario settings in each scenario
C. Incomplete executions should be avoided because they only give visibility into better scenario design
D. When resolving an incomplete execution, the scenario will process the bundle of data through the remainder of the scenario not yet executed

Correct Answer: B, D
Section:
Explanation:
B . Incomplete Executions Are Not Stored by Default:
By default, Workfront Fusion does not store incomplete executions unless this setting is explicitly enabled in the scenario's settings.
This feature needs to be activated for troubleshooting and manually resolving incomplete executions later.
D . Resolving Incomplete Executions:
When an incomplete execution is resumed, the scenario picks up from the point of failure and processes the remaining steps with the data bundle that caused the error.
This ensures that the scenario completes its intended process without restarting entirely.
Why Not Other Options?
A . When incomplete executions are togged, a user must review the History tab to understand the error and then manually run the entire scenario to resolve: This is incorrect because incomplete executions do not require restarting the entire scenario; they continue from the point of failure.
C . Incomplete executions should be avoided because they only give visibility into better scenario design: This statement is misleading. Incomplete executions are often inevitable and provide critical insights into handling errors and improving scenarios.
Adobe Workfront Fusion Documentation: Working with Incomplete Executions
Experience League Community: How to Enable and Manage Incomplete Executions in Fusion

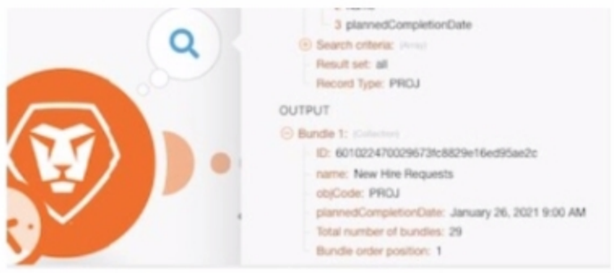
QUESTION 22
A Fusion user notices that a third-party web service is sometimes returning a connection error -'
'service is not reachable'. However, the module executes successfully a few minutes later in a new execution.

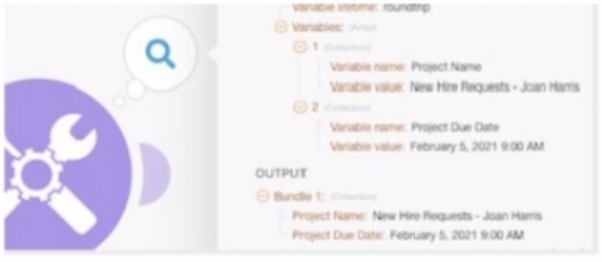
Which action increases the success rate of the executions?

A. Adding an error handler that will notify the system owner
B. Making use of the default error handling
C. Adding a Break directive to the module

Correct Answer: B
Section:
Explanation:
When dealing with intermittent errors, such as 'service is not reachable,' the default error handling in Adobe Workfront Fusion is often sufficient to improve execution success rates.
Default Error Handling:
Fusion automatically retries operations when transient errors, such as network or connection issues, occur.
By leveraging this built-in functionality, the system will attempt to re-execute the failing module after a brief delay, which is often enough for the external service to become reachable again.
Why Not Other Options?
A . Adding an error handler that will notify the system owner: While notifying the owner can be useful for monitoring purposes, it does not directly address improving the success rate of executions.
C . Adding a Break directive to the module: Adding a Break directive will stop the execution entirely, which is counterproductive in this case, as the service typically becomes reachable again after a short time.
Adobe Workfront Fusion Documentation: Default Error Handling Mechanisms
Experience League Community: Managing Intermittent API Errors in Fusion

QUESTION 23
What information can be understood from the images of this bundle inspector?





A. The trigger module returned 1 bundle of project data with only the project ID, name, and planned completion date getting passed to the second module in the scenario
B. The trigger module returned 29 bundles of task data with only the task ID. name, and planned completion date getting passed to the second module in the scenario
C. The first module delivered 29 bundles of data to the second module. Data manipulations happened in the second module to change the project name and planned completion date
D. The project name and planned completion date were changed in the trigger module because the input and output fields in the second module are the same

Correct Answer: C
Section:
Explanation:
Understanding the Scenario:
The scenario includes two modules:
Workfront Search Module: Fetches data (likely project details).
Tools Module: Manipulates and sets multiple variables.
What the Images Show:
Workfront Search Module (First Module):
The output indicates 29 bundles of project data were retrieved.
Each bundle contains fields such as ID, name, and plannedCompletionDate.
Tools Module (Second Module):
The data from the Workfront module is processed to set variables like Project Name and Project Due Date.
Output includes transformed values, such as appending the approver's name ('Joan Harris') to the project name and updating the project due date.
Why Option C is Correct:
Delivery of 29 Bundles: The Workfront module retrieved 29 project bundles and passed them to the Tools module.
Data Manipulation in Tools Module: The second module modified the data, such as appending 'Joan Harris' to the project name and updating the due date to a different value. This is evident from the changes in the output of the Tools module compared to the input from the Workfront module.
Why the Other Options are Incorrect:
Option A:
Incorrect because the Workfront module retrieved 29 bundles, not just 1 bundle. Additionally, the project name and planned completion date were manipulated in the second module, not the first.
Option B:
Incorrect because the retrieved data pertains to projects, not tasks.
Option D:
Incorrect because the input and output fields in the Tools module are not identical. The data was clearly manipulated within the Tools module.
How This Scenario Operates:
The Workfront module fetches raw data, which is then processed in the Tools module. The processed variables (e.g., modified project names and due dates) are prepared for further use in subsequent scenario steps.
Reference and Supporting Documentation:
Adobe Workfront Fusion Documentation: Bundle Inspector
Workfront Community: Using Tools Module for Data Manipulation
The correct interpretation is that the first module delivered 29 bundles to the second module, where data manipulations occurred to modify the project name and planned completion date.