100% Pass 2026 Workday Workday-Pro-Integrations: Workday Pro Integrations Certification Exam Practice Test

P.S. Free 2026 Workday Workday-Pro-Integrations dumps are available on Google Drive shared by Dumpexams: https://drive.google.com/open?id=1EXTL0_Kc2wyeIXTHlmWThlWVOf918ymF

Up to now, we have business connection with tens of thousands of exam candidates who adore the quality of our Workday-Pro-Integrations exam questions. Besides, we try to keep our services brief, specific and courteous with reasonable prices of Workday-Pro-Integrations Study Guide. All your questions will be treated and answered fully and promptly. So as long as you contact us to ask for the questions on the Workday-Pro-Integrations learning guide, you will get the guidance immediately.

If you have problems with your installation or use on our Workday-Pro-Integrations training guide, our 24 - hour online customer service will resolve your trouble in a timely manner. We dare say that our Workday-Pro-Integrations preparation quiz have enough sincerity to our customers. You can free download the demos of our Workday-Pro-Integrations Exam Questions which present the quality and the validity of the study materials and check which version to buy as well.

>> Workday-Pro-Integrations Practice Test <<

New Workday-Pro-Integrations Learning Materials, Workday-Pro-Integrations Latest Exam Price

The quality of our Workday-Pro-Integrations exam questions is of course in line with the standards of various countries. At the same time, our global market is also convenient for us to collect information. You will find that the update of Workday-Pro-Integrations learning quiz is very fast. You don't have to buy all sorts of information in order to learn more. Workday-Pro-Integrations training materials can meet all your needs. What are you waiting for? Just rush to buy them!

Workday Workday-Pro-Integrations Exam Syllabus Topics:

Topic	Details

Topic 1	• Calculated Fields: This section of the exam measures the skills of Workday Integration Analysts and covers the creation, configuration, and management of calculated fields used to transform, manipulate, and format data in Workday integrations. It evaluates understanding of field types, dependencies, and logical operations that enable dynamic data customization within integration workflows.
Topic 2	• Enterprise Interface Builders: This section of the exam measures the skills of Integration Developers and covers the use of Workday's Enterprise Interface Builder (EIB) to design, deploy, and maintain inbound and outbound integrations. It evaluates the candidate's ability to create templates, configure transformation rules, schedule integrations, and troubleshoot EIB workflows efficiently.
Topic 3	• Reporting: This section of the exam measures the skills of Reporting Analysts and focuses on building, modifying, and managing Workday reports that support integrations. It includes working with report writer tools, custom report types, calculated fields within reports, and optimizing report performance to support automated data exchange.
Topic 4	• Integrations: This section of the exam measures the skills of Integration Specialists and covers the full spectrum of integration techniques in Workday. It includes an understanding of core integration architecture, APIs, Workday Studio, and integration system user setup. The focus is on building scalable, maintainable, and secure integrations that ensure seamless system interoperability.

Workday Pro Integrations Certification Exam Sample Questions (Q12-Q17):

NEW QUESTION # 12
What is the purpose of granting an ISU modify access to the Integration Event domain via an ISSG?

- A. To let the ISU configure integration attributes and maps.
- B. To have the ISU own the integration schedule.
- C. To build the integration system as the ISU.
- D. To log into the user interface as the ISU and launch the integration.

Answer: A

Explanation:
Understanding ISUs and Integration Systems in Workday
* Integration System User (ISU):An ISU is a specialized user account in Workday designed for integrations, functioning as a service account to authenticate and execute integration processes. ISUs are created using the "Create Integration System User" task and are typically configured with settings like disabling UI sessions and setting long session timeouts (e.g., 0 minutes) to prevent expiration during automated processes. ISUs are not human users but are instead programmatic accounts used for API calls, EIBs, Core Connectors, or other integration mechanisms.
* Integration Systems:In Workday, an "integration system" refers to the configuration or setup of an integration, such as an External Integration Business (EIB), Core Connector, or custom integration via web services. Integration systems are defined to handle data exchange between Workday and external systems, and they require authentication, often via an ISU, to execute tasks like data retrieval, transformation, or posting.
* Assigning ISUs to Integration Systems:ISUs are used to authenticate and authorize integration systems to interact with Workday. When configuring an integration system, you assign an ISU to provide the credentials needed for the integration to run. This assignment ensures that theintegration can access Workday data and functionalities based on the security permissions granted to the ISU via its associated Integration System Security Group (ISSG).
* Limitation on Assignment:Workday's security model imposes restrictions to maintain control and auditability. Specifically, an ISU is designed to be tied to a single integration system to ensure clear accountability, prevent conflicts, and simplify security management. This limitation prevents an ISU from being reused across multiple unrelated integration systems, reducing the risk of unintended access or data leakage.
Evaluating Each Option
Let's assess each option based on Workday's integration and security practices:
Option A: An ISU can be assigned to five integration systems.
* Analysis:This is incorrect. Workday does not impose a specific numerical limit like "five" for ISU assignments to integration systems. Instead, the limitation is more restrictive: an ISU is typically assigned to only one integration system to ensure focused security and accountability. Allowing an ISU to serve multiple systems could lead to confusion, overlapping permissions, or security risks, which Workday's design avoids.
* Why It Doesn't Fit:There's no documentation or standard practice in Workday Pro Integrations suggesting a limit of five integration

systems per ISU. This option is arbitrary and inconsistent with Workday's security model.
Option B: An ISU can be assigned to an unlimited number of integration systems.
* Analysis:This is incorrect. Workday's security best practices do not allow an ISU to be assigned to an unlimited number of integration systems. Allowing this would create security vulnerabilities, as an ISU' s permissions (via its ISSG) could be applied across multiple unrelated systems, potentially leading to unauthorized access or data conflicts. Workday enforces a one-to-one or tightly controlled relationship to maintain auditability and security.
* Why It Doesn't Fit:The principle of least privilege and clear accountability in Workday integrations requires limiting an ISU's scope, not allowing unlimited assignments.
Option C: An ISU can be assigned to only one integration system.
* Analysis:This is correct. In Workday, an ISU is typically assigned to a single integration system to ensure that its credentials and permissions are tightly scoped. This aligns with Workday's security model, where ISUs are created for specific integration purposes (e.g., an EIB, Core Connector, or web service integration). When configuring an integration system, you specify the ISU in the integration setup (e.g., under "Integration System Attributes" or "Authentication" settings), and it is not reused across multiple systems to prevent conflicts or unintended access. This limitation ensures traceability and security, as the ISU's actions can be audited within the context of that single integration.
* Why It Fits:Workday documentation and best practices, including training materials and community forums, emphasize that ISUs are dedicated to specific integrations. For example, when creating an EIB or Core Connector, you assign an ISU, and it is not shared across other integrations unless explicitly reconfigured, which is rare and discouraged for security reasons.
Option D: An ISU can only be assigned to an ISSG and not an integration system.
* Analysis:This is incorrect. While ISUs are indeed assigned to ISSGs to inherit security permissions (as established in Question 26), they are also assigned to integration systems to provide authentication and authorization for executing integration tasks. The ISU's role includes both: it belongs to an ISSG for permissions and is linked to an integration system for execution. Saying it can only be assigned to an ISSG and not an integration system misrepresents Workday's design, as ISUs are explicitly configured in integration systems (e.g., EIB, Core Connector) to run processes.
* Why It Doesn't Fit:ISUs are integral to integration systems, providing credentials for API calls or data exchange. Excluding assignment to integration systems contradicts Workday's integration framework.
Final Verification
The correct answer is Option C, as Workday limits an ISU to a single integration system to ensure security, accountability, and clarity in integration operations. This aligns with the principle of least privilege, where ISUs are scoped narrowly to avoid overexposure. For example, when setting up a Core Connector: Job Postings (as in Question 25), you assign an ISU specifically for that integration, not multiple ones, unless reconfiguring for a different purpose, which is atypical.
Supporting Documentation
The reasoning is based on Workday Pro Integrations security practices, including:
* Workday Community documentation on creating and managing ISUs and integration systems.
* Tutorials on configuring EIBs, Core Connectors, and web services, which show assigning ISUs to specific integrations (e.g.,Workday Advanced Studio Tutorial).
* Integration security overviews from implementation partners (e.g., NetIQ, Microsoft Learn, Reco.ai) emphasizing one ISU per integration for security.
* Community discussions on Reddit and Workday forums reinforcing that ISUs are tied to single integrations for auditability (r/workday on Reddit).
This question focuses on the purpose of granting an Integration System User (ISU) modify access to the Integration Event domain via an Integration System Security Group (ISSG) in Workday Pro Integrations. Let' s analyze the role of the ISU, the Integration Event domain, and evaluate each option to determine the correct answer.
Understanding ISUs, ISSGs, and the Integration Event Domain
* Integration System User (ISU):As described in previous questions, an ISU is a service account for integrations, used to authenticate and execute integration processes in Workday. ISUs are assigned to ISSGs to inherit security permissions and are linked to specific integration systems (e.g., EIBs, Core Connectors) for execution.
* Integration System Security Group (ISSG):An ISSG is a security group that defines the permissions for ISUs, controlling what data and functionalities they can access or modify. ISSGs can be unconstrained (access all instances) or constrained (access specific instances based on context).
Permissions are granted via domain security policies, such as "Get," "Put," "View," or "Modify," applied to Workday domains.
* Integration Event Domain:In Workday, the Integration Event domain (or Integration Events security domain) governs access to integration-related activities, such as managing integration events, schedules, attributes, mappings, and logs. This domain is critical for integrations, as it controls the ability to create, modify, or view integration configurations and runtime events.
* "Modify" access to the Integration Event domain allows the ISU to make changes to integration configurations, such as attributes (e.g., file names, endpoints), mappings (e.g., data transformations), and event settings (e.g., schedules or triggers).
* This domain does not typically grant UI access or ownership of schedules but focuses on configuration and runtime control.
* Purpose of Granting Modify Access:Granting an ISU modify access to the Integration Event domain via an ISSG enables the ISU to perform configuration tasks for integrations, ensuring the integration system can adapt or update its settings programmatically. This is essential for automated integrations that need to adjust mappings, attributes, or event triggers without manual intervention. However, ISUs are not designed for UI interaction or administrative ownership, as they are service accounts.

Evaluating Each Option
Let's assess each option based on Workday's security and integration model:
Option A: To have the ISU own the integration schedule.
* Analysis:This is incorrect. ISUs do not "own" integration schedules or any other integration components. Ownership is not a concept applicable to ISUs, which are service accounts for execution, not administrative entities. Integration schedules are configured within the integration system (e.g., EIB or Core Connector) and managed by administrators or users with appropriate security roles, not by ISUs. Modify access to the Integration Event domain allows changes to schedules, but it doesn't imply ownership.
* Why It Doesn't Fit:ISUs lack administrative control or ownership; they execute based on permissions, not manage schedules as owners. This misinterprets the ISU's role.
Option B: To let the ISU configure integration attributes and maps.
* Analysis:This is correct. Granting modify access to the Integration Event domain allows the ISU to alter integration configurations, including attributes (e.g., file names, endpoints, timeouts) and mappings (e.g., data transformations like worker subtype mappings from Question 25). The Integration Event domain governs these configuration elements, and "Modify" permission enables the ISU to update them programmatically during integration execution. This is a standard use case for ISUs in automated integrations, ensuring flexibility without manual intervention.
* Why It Fits:Workday's documentation and training materials indicate that the Integration Event domain controls integration configuration tasks. For example, in an EIB or Core Connector, an ISU with modify access can adjust mappings or attributes, as seen in tutorials on integration setup (Workday Advanced Studio Tutorial). This aligns with the ISU's role as a service account for dynamic configuration.
Option C: To log into the user interface as the ISU and launch the integration.
* Analysis:This is incorrect. ISUs are not intended for UI interaction. When creating an ISU, a best practice is to disable UI sessions (e.g., set "Allow UI Sessions" to "No") and configure a session timeout of 0 minutes to prevent expiration during automation. ISUs operate programmaticallyvia APIs or integration systems, not through the Workday UI. Modify access to the Integration Event domain enables configuration changes, not UI login or manual launching.
* Why It Doesn't Fit:Logging into the UI contradicts ISU design, as they are service accounts, not user accounts. This option misrepresents their purpose.
Option D: To build the integration system as the ISU.
* Analysis:This is incorrect. ISUs do not "build" integration systems; they execute or configure existing integrations based on permissions. Building an integration system (e.g., creating EIBs, Core Connectors, or web services) is an administrative task performed by users with appropriate security roles (e.g., Integration Build domain access), not ISUs. Modify access to the Integration Event domain allows configuration changes, not the creation or design of integration systems.
* Why It Doesn't Fit:ISUs lack the authority or capability to build integrations; they are for runtime execution and configuration, not development or design.
Final Verification
The correct answer is Option B, as granting an ISU modify access to the Integration Event domain via an ISSG enables it to configure integration attributes (e.g., file names, endpoints) and maps (e.g., data transformations), which are critical for dynamic integration operations. This aligns with Workday's security model, where ISUs handle automated tasks within defined permissions, not UI interaction, ownership, or system building.
For example, in the Core Connector: Job Postings from Question 25, an ISU with modify access to Integration Event could update the filename pattern or worker subtype mappings, ensuring the integration adapts to vendor requirements without manual intervention. This is consistent with Workday's design for integration automation.
Supporting Documentation
The reasoning is based on Workday Pro Integrations security practices, including:
* Workday Community documentation on ISUs, ISSGs, and domain security (e.g., Integration Event domain permissions).
* Tutorials on configuring EIBs and Core Connectors, showing ISUs modifying attributes and mappings (Workday Advanced Studio Tutorial).
* Integration security overviews from implementation partners (e.g., NetIQ, Microsoft Learn, Reco.ai) detailing domain access for ISUs.
* Community discussions on Reddit and Workday forums reinforcing ISU roles for configuration, not UI or ownership (r/workday on Reddit).

NEW QUESTION # 13
A vendor needs to create a Date Difference calculated field. However, the two dates needed for that calculation are on two separate business objects.
What additional calculated field do you need to create that Date Difference calculated field?

- A. Lookup Value as of Date
- B. Lookup Related Value

- C. Build Date
- D. Lookup Date Rollup

Answer: B

Explanation:
When creating a Date Difference calculated field in Workday, both dates must exist on the same business object. If they are on different business objects, you need to first bring the second date onto the primary object. To do that, you use a:
Lookup Related Value calculated field - this allows you to retrieve a field (like a date) from a related business object, so it can then be used in further calculations.
Example scenario:
You want to subtract Hire Date (on the Worker object) from Dependent's Birth Date (on the Dependent object).
These are on different objects → use Lookup Related Value to pull the second date into the current object context.
Then, create the Date Difference using both dates on the same object.
Why other options are incorrect:
B . Build Date creates a synthetic date, not for bridging objects.
C . Lookup Date Rollup rolls up values across multiple related objects, not typically used for 1-to-1 value bridging.
D . Lookup Value as of Date is used for time-sensitive lookups (e.g., point-in-time values), not structural bridging.

NEW QUESTION # 14
What is the workflow to upload an XSLT file for a brand new Document Transformation system?

- A. Create Integration Attachment Service, then Configure Integration Attachment Service
- B. Configure XSLT Attachment Transformation, then Create Integration Attachment Service
- C. Create XSLT Attachment Transformation, then Configure Integration Attachment Service
- D. Configure Integration Attachment Service, then Create Integration Service Attachment

Answer: C

Explanation:
In the Workday Pro Integrations program, the process of uploading an XSLT file for a brand-new Document Transformation system follows a specific workflow designed to ensure the transformation logic is properly attached and configured within the integration system. The correct sequence involves first creating the XSLT Attachment Transformation and then configuring the Integration Attachment Service to utilize it. Here's a step-by-step breakdown based on Workday's integration methodology:
Create XSLT Attachment Transformation:
The initial step is to create an XSLT Attachment Transformation object within Workday. This involves uploading the XSLT file, which contains the transformation logic needed to convert XML data into the desired format for the Document Transformation system. In Workday, XSLT (Extensible Stylesheet Language Transformations) is used to define how data from a source (typically in XML format) is transformed into an output format compatible with an external system.
To do this, you navigate to the Integration System, access the related actions, and select the option to create a new "XSLT Attachment Transformation." You then name the transformation, upload the XSLT file (with a size limit of 30 MB as per Workday specifications), and save it. This step establishes the transformation logic as an object that can be referenced by the integration system.
Configure Integration Attachment Service:
Once the XSLT Attachment Transformation is created, the next step is to configure the Integration Attachment Service to incorporate this transformation. The Integration Attachment Service is a component of the Document Transformation system that handles the delivery or processing of the transformed data.
In this step, you edit the integration system, navigate to the "Services" tab, and configure the Integration Attachment Service. Here, you specify the previously created XSLT Attachment Transformation as the transformation to be applied. This links the XSLT logic to the integration workflow, ensuring that the data processed by the Document Transformation system is transformed according to the uploaded XSLT file.
Why Other Options Are Incorrect:
A . Configure XSLT Attachment Transformation, then Create Integration Attachment Service: This is incorrect because you cannot "configure" an XSLT Attachment Transformation before it exists. It must first be created as an object in Workday before any configuration or association with services can occur.
C . Create Integration Attachment Service, then Configure Integration Attachment Service: This option skips the creation of the XSLT Attachment Transformation entirely, which is a critical step. Without the transformation defined, configuring the service alone would not enable the XSLT upload or its functionality.
D . Configure Integration Attachment Service, then Create Integration Service Attachment: This sequence is reversed and misleading. The Integration Attachment Service must be configured to use an existing XSLT Attachment Transformation, not the

other way around. Additionally, "Create Integration Service Attachment" is not a standard term in this context within Workday documentation.
Workday Pro Integrations Study Guide Reference:
Workday Integration System Fundamentals: This section outlines the components of an integration system, including the use of XSLT for document transformation and the role of attachment services.
Document Transformation Module: Specifically details the process of uploading and applying XSLT files, emphasizing the creation of an XSLT Attachment Transformation followed by its configuration within the integration services.
Core Connectors and Document Transformation Course Manual: Provides practical steps for setting up transformations, including the sequence of creating and then configuring transformation attachments (e.g., Activities related to "Upload a Custom XSLT Transformation" and "Edit XSLT Attachment Transformation").
Workday Community Documentation: Confirms that XSLT files are uploaded as attachment transformations and then linked to services like the Integration Attachment Service for processing.

NEW QUESTION # 15
Refer to the following scenario to answer the question below.
You have configured a Core Connector: Worker integration, which utilizes the following basic configuration:
* Integration field attributes are configured to output the Position Title and Business Title fields from the Position Data section.
* Integration Population Eligibility uses the field Is Manager which returns true if the worker holds a manager role.
* Transaction Log service has been configured to Subscribe to specific Transaction Types: Position Edit Event.
You launch your integration with the following date launch parameters (Date format of MM/DD/YYYY):
* As of Entry Moment: 05/25/2024 12:00:00 AM * Effective Date: 05/25/2024
* Last Successful As of Entry Moment: 05/23/2024 12:00:00 AM
* Last Successful Effective Date: 05/23/2024
To test your integration, you made a change to a worker named Jared Ellis who is assigned to the manager role for the IT Help Desk department. You use the Change Business Title related action on Jared and update the Business Title of the position to a new value. Jared Ellis' worker history shows the Title Change Event as being successfully completed with an effective date of 05/24/2024 and an Entry Moment of 05/24/2024 07:58:
53 AM however Jared Ellis does not show up in your output. What configuration element would have to be modified for the integration to include Jared Ellis in the output?

- A. Integration Population Eligibility
- B. Integration Field Attributes
- C. Transaction log subscription
- D. Date launch parameters

Answer: C

Explanation:
The scenario involves a Core Connector: Worker integration configured to output Position Title and Business Title fields for workers who meet the Integration Population Eligibility criteria (Is Manager = true), with the Transaction Log service subscribed to the "Position Edit Event." The integration is launched with specific date parameters, and a test is performed by updating Jared Ellis' Business Title using the "Change Business Title" related action. Jared is a manager, and the change is logged with an effective date of 05/24/2024 and an entry moment of 05/24/2024 07:58:53 AM. Despite this, Jared does not appear in the output. Let's determine why and identify the configuration element that needs modification.
In Workday, the Core Connector: Worker integration uses the Transaction Log service to detect changes based on subscribed transaction types. The subscribed transaction type in this case is "Position Edit Event," which is triggered when a position is edited via the "Edit Position" business process. However, the test scenario involves a "Change Business Title" related action, which is a distinct business process in Workday.
This action updates the Business Title field but does not necessarily trigger a "Position Edit Event." Instead, it generates a different event type, such as a "Title Change Event" (as noted in Jared's worker history), depending on how the system logs the action.
The date launch parameters provided are:
* As of Entry Moment:05/25/2024 12:00:00 AM - The latest point for entry moments.
* Effective Date:05/25/2024 - The latest effective date for changes.
* Last Successful As of Entry Moment:05/23/2024 12:00:00 AM - The starting point for entry moments from the last run.
* Last Successful Effective Date:05/23/2024 - The starting point for effective dates from the last run.
Jared's change has:
* Entry Moment:05/24/2024 07:58:53 AM - Falls between 05/23/2024 12:00:00 AM and 05/25/2024 12:
00:00 AM.
* Effective Date:05/24/2024 - Falls between 05/23/2024 and 05/25/2024.
The date parameters correctly cover the time window of Jared's change, meaning the issue is not with the date range but with the

event detection logic. The Transaction Log subscription determines which events are processed by the integration. Since the subscription is set to "Position Edit Event" and the change was made via "Change Business Title" (logged as a "Title Change Event"), the integration does not recognize this event because it is not subscribed to the appropriate transaction type.
To include Jared Ellis in the output, theTransaction Log subscriptionmust be modified to include the event type associated with the "Change Business Title" action, such as "Title Change Event" or a broader category like "Position Related Event" that encompasses both position edits and title changes. This ensures the integration captures the specific update made to Jared's Business Title.
Let's evaluate the other options:
* B. Date launch parameters:The parameters already include Jared's entry moment and effective date within the specified ranges (05/23/2024 to 05/25/2024). Adjusting these would not address the mismatch between the subscribed event type and the actual event triggered.
* C. Integration Field Attributes:These are set to output Position Title and Business Title, and the change to Business Title is within scope. The field configuration is correct and does not need modification.
* D. Integration Population Eligibility:This is set to "Is Manager = true," and Jared is a manager. This filter is functioning as intended and is not the issue.
The root cause is the Transaction Log subscription not aligning with the event type generated by the "Change Business Title" action, makingA. Transaction log subscriptionthe correct answer.
Workday Pro Integrations Study Guide References
* Workday Integrations Study Guide: Core Connector: Worker- Section on "Transaction Log Configuration" explains how subscribing to specific transaction types filters the events processed by the integration.
* Workday Integrations Study Guide: Change Detection- Details how different business processes (e.g., Edit Position vs. Change Business Title) generate distinct event types in the Transaction Log.
* Workday Integrations Study Guide: Event Subscription- Notes the importance of aligning subscription types with the specific business actions being tested or monitored.

NEW QUESTION # 16
What is the workflow to upload an XSLT file for a brand new Document Transformation system?

- A. Create Integration Attachment Service, then Configure Integration Attachment Service
- B. Configure XSLT Attachment Transformation, then Create Integration Attachment Service
- C. Create XSLT Attachment Transformation, then Configure Integration Attachment Service
- D. Configure Integration Attachment Service, then Create Integration Service Attachment

Answer: C

Explanation:
In the Workday Pro Integrations program, the process of uploading an XSLT file for a brand-new Document Transformation system follows a specific workflow designed to ensure the transformation logic is properly attached and configured within the integration system. The correct sequence involves first creating the XSLT Attachment Transformation and then configuring the Integration Attachment Service to utilize it. Here's a step-by-step breakdown based on Workday's integration methodology:
* Create XSLT Attachment Transformation:
* The initial step is to create an XSLT Attachment Transformation object within Workday. This involves uploading the XSLT file, which contains the transformation logic needed to convert XML data into the desired format for the Document Transformation system. In Workday, XSLT (Extensible Stylesheet Language Transformations) is used to define how data from a source (typically in XML format) is transformed into an output format compatible with an external system.
* To do this, you navigate to the Integration System, access the related actions, and select the option to create a new "XSLT Attachment Transformation." You then name the transformation, upload the XSLT file (with a size limit of 30 MB as per Workday specifications), and save it.
This step establishes the transformation logic as an object that can be referenced by the integration system.
* Configure Integration Attachment Service:
* Once the XSLT Attachment Transformation is created, the next step is to configure the Integration Attachment Service to incorporate this transformation. The Integration Attachment Service is a component of the Document Transformation system that handles the delivery or processing of the transformed data.
* In this step, you edit the integration system, navigate to the "Services" tab, and configure the Integration Attachment Service. Here, you specify the previously created XSLT Attachment Transformation as the transformation to be applied. This links the XSLT logic to the integration workflow, ensuring that the data processed by the Document Transformation system is transformed according to the uploaded XSLT file.
Why Other Options Are Incorrect:
* A. Configure XSLT Attachment Transformation, then Create Integration Attachment Service:
This is incorrect because you cannot "configure" an XSLT Attachment Transformation before it exists.
It must first be created as an object in Workday before any configuration or association with services can occur.

* C. Create Integration Attachment Service, then Configure Integration Attachment Service: This option skips the creation of the XSLT Attachment Transformation entirely, which is a critical step.
Without the transformation defined, configuring the service alone would not enable the XSLT upload or its functionality.
* D. Configure Integration Attachment Service, then Create Integration Service Attachment: This sequence is reversed and misleading. The Integration Attachment Service must be configured to use an existing XSLT Attachment Transformation, not the other way around. Additionally, "Create Integration Service Attachment" is not a standard term in this context within Workday documentation.
Workday Pro Integrations Study Guide References:
* Workday Integration System Fundamentals: This section outlines the components of an integration system, including the use of XSLT for document transformation and the role of attachment services.
* Document Transformation Module: Specifically details the process of uploading and applying XSLT files, emphasizing the creation of an XSLT Attachment Transformation followed by its configuration within the integration services.
* Core Connectors and Document Transformation Course Manual: Provides practical steps for setting up transformations, including the sequence of creating and then configuring transformation attachments (e.g., Activities related to "Upload a Custom XSLT Transformation" and "Edit XSLT Attachment Transformation").
* Workday Community Documentation: Confirms that XSLT files are uploaded as attachment transformations and then linked to services like the Integration Attachment Service for processing.

NEW QUESTION # 17
.....