Workday-Pro-Integrations試験の準備方法｜信頼的なWorkday-Pro-Integrations試験関連情報試験｜有効的なWorkday Pro Integrations Certification Exam受験体験

2025年JPTestKingの最新Workday-Pro-Integrations PDFダンプおよびWorkday-Pro-Integrations試験エンジンの無料共有：https://drive.google.com/open?id=1DQ7zWYUYLkYX6lwjLM1mJ_-5mgcwLPf4

JPTestKingは、これまでで最高のWorkday-Pro-Integrations学習ガイドだけでなく、最も効率的な顧客のサーバーも提供することで、この分野で最も人気のある評判を所有しています。 Workday-Pro-Integrations認定資格を取得し、希望するより高い給与を達成するための最善かつ最速の方法をご案内します。 Workday-Pro-Integrations試験の準備により、Workday-Pro-Integrations学習質問の成績が向上し、生活の状態を変えることができます。専門的な知識の蓄積です。 Workday-Pro-Integrationsブレインダンプでより成功するでしょう。

短時間で試験に合格して認定資格を取得する場合は、適切なWorkday-Pro-Integrations試験問題を選択することが非常に重要です。 Workday-Pro-Integrations学習資料にもっと注意を払う必要があります。 。すべてのお客様に適切な学習教材を提供するために、当社の多くの専門家がWorkday-Pro-Integrationsトレーニング教材を設計しました。 Workday-Pro-Integrations試験の質問を購入すると、Workday-Pro-Integrations試験に合格して認定資格を取得するのが非常に簡単になると約束できます。

>> Workday-Pro-Integrations試験関連情報 <<

有難いWorkday-Pro-Integrations試験関連情報と更新するWorkday-Pro-Integrations受験体験

Workday-Pro-Integrations学習教材は、試験にすばやく合格し、希望する証明書を取得するのに役立ちます。その後、あなたは良い仕事を得るためにもう一つのチップを持っています。 Workday-Pro-Integrations学習教材を使用すると、より高い出発点に立って、Workday-Pro-Integrations試験に他の人よりも一歩早く合格し、他の人よりも早くチャンスを活用できます。このペースの速い社会では、あなたの時間はとても貴重です。 1人の力だけに頼る場合、あなたが優位に立つことは困難です。 Workday-Pro-Integrationsの学習に関する質問は、最も満足のいくアシスタントになります。

Workday Workday-Pro-Integrations 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• Cloud Connect：このセクションでは、Workday導入コンサルタントのスキルを評価し、サードパーティとの統合におけるWorkday Cloud Connectソリューションの活用に焦点を当てます。構築済みのコネクタ、構成設定、そしてセキュリティとデータの整合性を確保しながらWorkdayと外部システム間のデータフローを管理する方法の理解が問われます。
トピック 2	• エンタープライズインターフェースビルダー：このセクションでは、統合開発者のスキルを評価し、Workdayのエンタープライズインターフェースビルダー（EIB）を使用してインバウンドおよびアウトバウンド統合を設計、展開、保守するスキルを網羅します。テンプレートの作成、変換ルールの設定、統合のスケジュール設定、EIBワークフローの効率的なトラブルシューティングを行う受験者の能力を評価します。
トピック 3	• XSLT：このセクションでは、データ統合開発者のスキルを評価し、Workday 統合における XSLT（Extensible Stylesheet Language Transformations）の使用について学習します。XML データ構造の変換、条件付きロジックの適用、API や外部ファイル配信などの様々な統合ユースケース向けの出力のフォーマット設定に重点が置かれます。

Workday Pro Integrations Certification Exam 認定 Workday-Pro-Integrations 試験問題 (Q47-Q52):

質問 # 47
What is the relationship between an ISU (Integration System User) and an ISSG (Integration System Security Group)?

- A. The ISU controls what accounts are in the ISSG.
- B. The ISU owns the ISSG.
- C. The ISU is a member of the ISSG.
- D. The ISU grants security policies to the ISSG.

正解：C

解説：
This question explores the relationship between an Integration System User (ISU) and an Integration System Security Group (ISSG) in Workday Pro Integrations, focusing on how security is structured for integrations.
Let's analyze the relationship and evaluate each option to determine the correct answer.
Understanding ISU and ISSG in Workday
* Integration System User (ISU):An ISU is a dedicated user account in Workday specifically designed for integrations. It acts as a "robot account" or service account, used by integration systems to interact with Workday via APIs, web services, or other integration mechanisms (e.g., EIBs, Core Connectors).
ISUs are typically configured with a username, password, and specific security settings, such as disabling UI sessions and setting session timeouts to prevent expiration (commonly set to 0 minutes).
ISUs are not human users but are instead programmatic accounts for automated processes.
* Integration System Security Group (ISSG):An ISSG is a security container or group in Workday that defines the permissions and access rights for integration systems. ISSGs are used to manage what data and functionalities an integration (or its associated ISU) can access or modify within Workday. There are two types of ISSGs:
* Unconstrained:Allows access to all data instances secured by the group.
* Constrained:Limits access to a subset of data instances based on context (e.g., specific segments or data scopes).ISSGs are configured with domain security policies, granting permissions like
"Get" (read), "Put" (write), "View," or "Modify" for specific domains (e.g., Worker Data, Integration Build).
* Relationship Between ISU and ISSG:In Workday, security for integrations is managed through a hierarchical structure. An ISU is associated with or assigned to an ISSG to inherit its permissions. The ISSG acts as the security policy container, defining what the ISU can do, while the ISU is the account executing those actions. This relationship ensures that integrations have controlled, audited access to Workday data and functions, adhering to the principle of least privilege.
Evaluating Each Option
Let's assess each option based on Workday's security model for integrations:
Option A: The ISU is a member of the ISSG.
* Analysis:This is correct. In Workday, an ISU is assigned to or associated with an ISSG to gain the necessary permissions. The

ISSG serves as a security group that contains one or more ISUs, granting them access to specific domains and functionalities. For example, when creating an ISU, you use the
"Create Integration System User" task, and then assign it to an ISSG via the "Assign Integration System Security Groups" or "Maintain Permissions for Security Group" tasks. Multiple ISUs can belong to the same ISSG, inheriting its permissions. This aligns with Workday's security framework, where security groups (like ISSGs) manage user (or ISU) access.
* Why It Fits:The ISU is a "member" of the ISSG in the sense that it is linked to the group to receive its permissions, enabling secure integration operations. This is a standard practice for managing integration security in Workday.
Option B: The ISU owns the ISSG.
* Analysis:This is incorrect. In Workday, ISUs do not "own" ISSGs. Ownership or control of security groups is not a concept applicable to ISUs, which are service accounts for integrations, not administrative entities with authority over security structures. ISSGs are created and managed by Workday administrators or security professionals using tasks like "Create Security Group" and "Maintain Permissions for Security Group." The ISU is simply a user account assigned to the ISSG, not its owner or controller.
* Why It Doesn't Fit:Ownership implies administrative control, which ISUs lack; they are designed for execution, not management of security groups.
Option C: The ISU grants security policies to the ISSG.
* Analysis:This is incorrect. ISUs do not have the authority to grant or modify security policies for ISSGs. Security policies are defined and assigned to ISSGs by Workday administrators or security roles with appropriate permissions (e.g., Security Configuration domain access). ISUs are passive accounts that execute integrations based on the permissions granted by the ISSG they are assigned to. Granting permissions is an administrative function, not an ISU capability.
* Why It Doesn't Fit:ISUs are integration accounts, not security administrators, so they cannot modify or grant policies to ISSGs.
Option D: The ISU controls what accounts are in the ISSG.
* Analysis:This is incorrect. ISUs do not control membership or configuration of ISSGs. Adding or removing accounts (including other ISUs) from an ISSG is an administrative task performed by users with security configuration permissions, using tasks like "Maintain Permissions for Security Group." ISUs are limited to executing integration tasks based on their assigned ISSG permissions, not managing group membership.
* Why It Doesn't Fit:ISUs lack the authority to manage ISSG membership or structure, as they are not administrative accounts but integration-specific service accounts.
Final Verification
Based on Workday's security model, the correct relationship is that an ISU is a member of an ISSG, inheriting its permissions to perform integration tasks. This is consistent with the principle of least privilege, where ISSGs define access, and ISUs execute within those boundaries. The other options misattribute administrative or ownership roles to ISUs, which are not supported by Workday's design.
Supporting Information
The relationship is grounded in Workday's integration security practices, including:
* Creating an ISU via the "Create Integration System User" task.
* Creating an ISSG via the "Create Security Group" task, selecting "Integration System Security Group (Unconstrained)" or "Constrained."
* Assigning the ISU to the ISSG using tasks like "Assign Integration System Security Groups" or
"Maintain Permissions for Security Group."
* Configuring domain security policies (e.g., Get, Put) for the ISSG to control ISU access to domains like Worker Data, Integration Build, etc.
* Activating security changes via "Activate Pending Security Policy Changes." This structure ensures secure, controlled access for integrations, with ISSGs acting as the permission container and ISUs as the executing accounts.
Key References
The explanation aligns with Workday Pro Integrations documentation and best practices, including:
* Integration security overviews and training on Workday Community.
* Guides for creating ISUs and ISSGs in implementation documentation (e.g., NetIQ, Microsoft Learn, Reco.ai).
* Tutorials on configuring domain permissions and security groups for integrations (e.g., ServiceNow, Apideck, Surety Systems).

質問 # 48
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. Transaction log subscription
- C. Integration Field Attributes
- D. Date launch parameters

正解： B

解説：
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, the Transaction Log subscription must 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, making A. Transaction log subscription the correct answer.
Workday Pro Integrations Study Guide Reference
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.

質問 # 49

Refer to the scenario. You are configuring a Core Connector: Worker integration with the Data Initialization Service (DIS) enabled, scheduled to run once daily. The integration must extract only active worker records with changes to compensation, home address, or business title since the last 24 hours. It uses Workday's change detection to avoid full extracts.

During testing, the Core Connector: Worker DIS output unexpectedly includes terminated workers, even though the change detection date parameters are correctly defined for a Full-Diff extract. The requirements specify that only active workers should be included in the output.

What configuration step should you modify to ensure the integration excludes terminated workers?

- A. Configure Integration Population Eligibility step to filter out terminated employees.
- B. Configure Integration Attributes for Integration System step to enable Include Inactive Workers in Full File.
- C. Configure Integration Field Overrides step to use the correct Eligibility Criterion to filter out terminated employees.
- D. Configure Integration Transaction Log step to subscribe to everything except termination transactions.

正解： A

解説：

This scenario addresses an issue where a Core Connector: Worker integration - with DIS enabled and Full- Diff mode configured - unexpectedly includes terminated workers in the output, despite a requirement to include only active workers.

The correct step to address this issue is the configuration of Integration Population Eligibility.

From the Workday Pro: Integrations - Core Connector Configuration Guide, the relevant extract states:

"The Integration Population Eligibility step allows users to define which workers or populations are eligible to be included in the integration output. This includes filtering by worker status, organization, supervisory org, or other eligibility criteria. If this is not configured to exclude terminated workers, the integration will include all workers who meet the event conditions, regardless of their current status." Even though the integration uses change detection and the correct launch parameters, Workday still considers any worker with a qualifying change, including those terminated, unless they are explicitly excluded via eligibility rules.

Therefore, to prevent terminated workers from appearing in the output, you must set a filter in the Integration Population Eligibility step to include only active workers (e.g., using Worker.Status = Active or similar criteria).

Incorrect Options Explained:

* A. Configure Integration Attributes... Include Inactive Workers in Full FileThis option would cause inactive (e.g., terminated) workers to be included when enabled. It doesn't help filter them out.

* B. Configure Integration Transaction Log... subscribe to everything except terminationSubscription controls which events trigger processing but does not control population eligibility. Terminated workers with address changes prior to termination could still appear if eligibility is not defined.

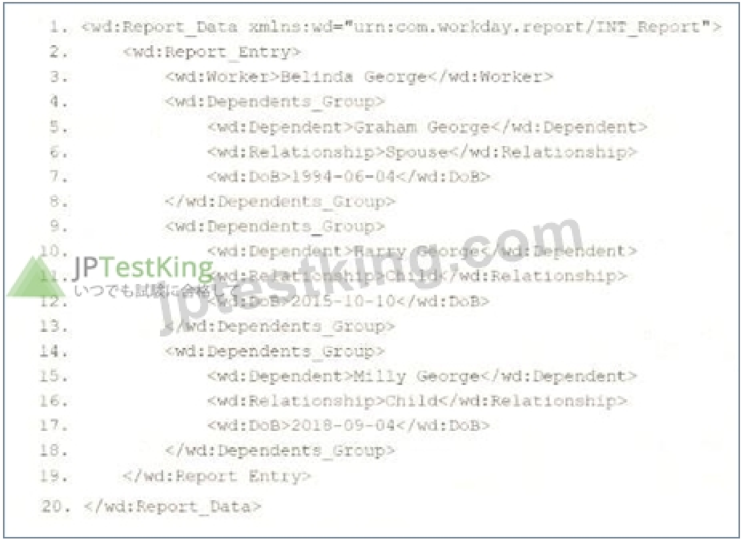
* D. Configure Integration Field Overrides... use Eligibility CriterionField Overrides change data mappings or formats, not population eligibility. It cannot exclude terminated workers.

References:

Workday Pro: Integrations Curriculum - Core Connector: Worker Configuration and Population Eligibility Workday Community: Integration System Configuration > Integration Population Eligibility Workday Training Materials: Core Connector Deployment Best Practices

質問 # 50

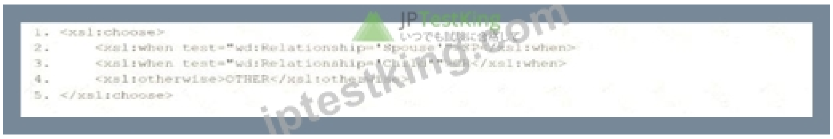
Refer to the following XML to answer the question below.



```
<wd:Report_Data xmlns:wd="urn:com.workday.report/INT_Report">
    <wd:Report_Entry>
        <wd:Worker>Belinda George</wd:Worker>
        <wd:Dependents_Group>
            <wd:Dependent>Graham George</wd:Dependent>
            <wd:Relationship>Spouse</wd:Relationship>
            <wd:DoB>1994-06-04</wd:DoB>
        </wd:Dependents_Group>
        <wd:Dependents_Group>
            <wd:Dependent>Harry George</wd:Dependent>
            <wd:Relationship>Child</wd:Relationship>
            <wd:DoB>2015-10-10</wd:DoB>
        </wd:Dependents_Group>
        <wd:Dependents_Group>
            <wd:Dependent>Milly George</wd:Dependent>
            <wd:Relationship>Child</wd:Relationship>
            <wd:DoB>2018-09-04</wd:DoB>
        </wd:Dependents_Group>
    </wd:Report Entry>
</wd:Report_Data>
```

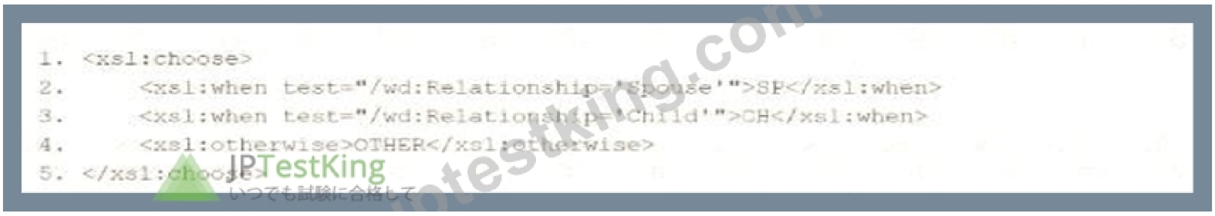
You are an integration developer and need to write XSLT to transform the output of an EIB which is using a web service enabled

report to output worker data along with their dependents. You currently have a template which matches on wd:Dependents_Group to iterate over each dependent. Within the template which matches on wd:Dependents_Group you would like to output a relationship code by using an <xsl:choose> statement.
What XSLT syntax would be used to output SP when the dependent relationship is spouse, output CH when the dependent relationship is child, otherwise output OTHER?

- A. 

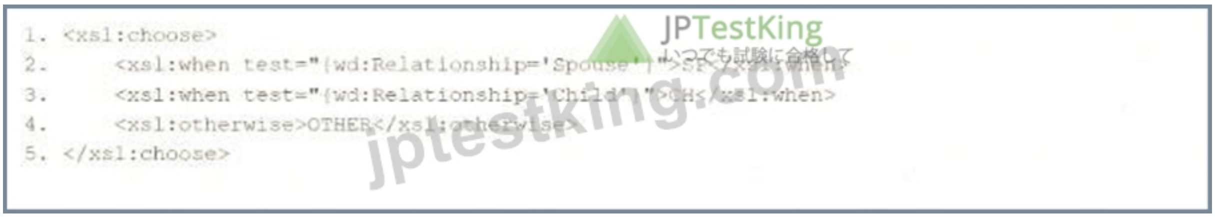
```
<xsl:choose>
    <xsl:when test="wd:Relationship='Spouse'">SP</xsl:when>
    <xsl:when test="wd:Relationship='Child'">CH</xsl:when>
    <xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

- B.

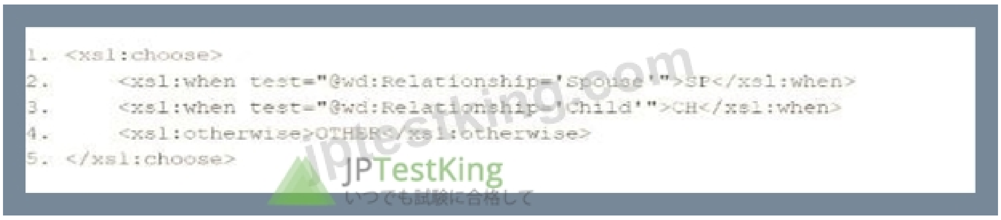


```
<xsl:choose>
    <xsl:when test="/wd:Relationship='Spouse'">SP</xsl:when>
    <xsl:when test="/wd:Relationship='Child'">CH</xsl:when>
    <xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

- C. B.



```
<xsl:choose>
    <xsl:when test="{wd:Relationship='Spouse'}">SP</xsl:when>
    <xsl:when test="{wd:Relationship='Child'}">CH</xsl:when>
    <xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```



```
<xsl:choose>
    <xsl:when test="@wd:Relationship='Spouse'">SP</xsl:when>
    <xsl:when test="@wd:Relationship='Child'">CH</xsl:when>
    <xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

- D.

正解： D

解説：
In Workday integrations, XSLT is used to transform XML data, such as the output from an Enterprise Interface Builder (EIB) or a web service-enabled report, into a desired format for third-party systems. In this scenario, you need to write XSLT to process wd:Dependents_Group elements and output a relationship code based on the value of the wd:Relationship attribute or element. The requirement is to output "SP" for a
"Spouse" relationship, "CH" for a "Child" relationship, and "OTHER" for any other relationship, using an <xsl:choose> statement within a template matching wd:Dependents_Group.
Here's why option C is correct:
* XSLT <xsl:choose> Structure: The <xsl:choose> element in XSLT provides conditional logic similar to a switch statement. It evaluates conditions in <xsl:when> elements sequentially, executing the first matching condition, and uses <xsl:otherwise> for any case that doesn't match.
* Relationship as an Attribute: Based on the provided XML snippet, wd:Relationship is an attribute (e.g., <wd:Relationship>Spouse</wd:Relationship> within wd:Dependents_Group). However, in Workday XML for integrations, wd:Relationship is often represented as an attribute (@wd:Relationship) rather than a child element, especially in contexts like dependent data in reports. The syntax @wd:
Relationship in the test attribute of <xsl:when> correctly references this attribute, aligning with Workday's typical XML structure for such data.
* Condition Matching:
* The first <xsl:when test="@wd:Relationship='Spouse'">SP</xsl:when> checks if the wd:
Relationship attribute equals "Spouse" and outputs "SP" if true.
* The second <xsl:when test="@wd:Relationship='Child'">CH</xsl:when> checks if the wd:
Relationship attribute equals "Child" and outputs "CH" if true.
* The <xsl:otherwise>OTHER</xsl:otherwise> handles all other cases, outputting "OTHER" if the relationship is neither "Spouse" nor "Child."
* Context in Template: Since the template matches on wd:Dependents_Group, the test conditions operate on the current wd:Dependents_Group element and its attributes, ensuring the correct relationship code is output for each dependent. The XML

snippet shows wd:Relationship as an element, but Workday documentation and integration practices often standardize it as an attribute in XSLT transformations, making @wd:Relationship appropriate.
Why not the other options?
* A.
xml
WrapCopy

```
<xsl:choose>
<xsl:when test="wd:Relationship='Spouse'">SP</xsl:when>
<xsl:when test="wd:Relationship='Child'">CH</xsl:when>
<xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

This assumes wd:Relationship is a child element of wd:Dependents_Group, not an attribute. The XML snippet shows wd:Relationship as an element, but in Workday integrations, XSLT often expects attributes for efficiency and consistency, especially in report outputs. Using wd:Relationship without @ would not match the attribute-based structure commonly used, making it incorrect for this context.
* B.
xml
WrapCopy

```
<xsl:choose>
<xsl:when test="@wd:Relationship='Spouse'">SP</xsl:when>
<xsl:when test="@wd:Relationship='Child'">CH</xsl:when>
<xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

This correctly uses @wd:Relationship for an attribute but has a logical flaw: if wd:Relationship='Child', the second <xsl:when> would output "CH," but the order of conditions matters. However, the primary issue is that it doesn't match the exact structure or intent as clearly as option C, and Workday documentation often specifies exact attribute-based conditions like those in option C.
* D.
xml
WrapCopy

```
<xsl:choose>
<xsl:when test="/wd:Relationship='Spouse'">SP</xsl:when>
<xsl:when test="/wd:Relationship='Child'">CH</xsl:when>
<xsl:otherwise>OTHER</xsl:otherwise>
</xsl:choose>
```

This uses an absolute path (/wd:Relationship), which searches for a wd:Relationship element at the root of the XML document, not within the current wd:Dependents_Group context. This would not work correctly for processing dependents in the context of the template matching wd:Dependents_Group, making it incorrect.
To implement this in XSLT:
* Within your template matching wd:Dependents_Group, you would include the <xsl:choose> statement from option C to evaluate the wd:Relationship attribute and output the appropriate relationship code ("SP," "CH," or "OTHER") based on its value. This ensures the transformation aligns with Workday's XML structure and integration requirements for processing dependent data in an EIB or web service- enabled report, even though the provided XML shows wd:Relationship as an element-XSLT transformations often normalize to attributes for consistency.
Workday Pro Integrations Study Guide: Section on "XSLT Transformations for Workday Integrations" - Details the use of <xsl:choose>, <xsl:when>, <xsl:otherwise>, and XPath for conditional logic in XSLT, including handling attributes like @wd:Relationship.
Workday EIB and Web Services Guide: Chapter on "XML and XSLT for Report Data" - Explains the structure of Workday XML (e.g., wd:Dependents_Group, @wd:Relationship) and how to use XSLT to transform dependent data, including attribute-based conditions.
Workday Reporting and Analytics Guide: Section on "Web Service-Enabled Reports" - Covers integrating report outputs with XSLT for transformations, including examples of conditional logic for relationship codes.

質問 # 51
What is the task used to upload a new XSLT file for a pre-existing document transformation integration system?

- A. Edit Integration Attachment Service
- B. Edit Integration Attachment
- C. Edit XSLT Attachment Transformation
- D. Edit Integration Service Attachment

正解：C

解説：
In Workday, when you need to upload a new XSLT (Extensible Stylesheet Language Transformations) file to modify or replace an existing transformation within a pre-existing document transformation integration system, the specific task required is "Edit XSLT Attachment Transformation." This task allows users to update the XSLT file that governs how XML data is transformed within the integration system without creating an entirely new transformation object.
Here's why this is the correct answer:
* Workday's integration systems often rely on XSLT to transform XML data into the desired format for downstream systems or processes. When an XSLT file has already been associated with an integration system (e.g., as part of an Enterprise Interface Builder (EIB) or a Document Transformation Connector), updating it requires accessing the existing transformation configuration.
* The "Edit XSLT Attachment Transformation" task enables users to upload a revised version of the XSLT file. This action replaces the previous file while maintaining the integration system's configuration, ensuring continuity without necessitating additional changes to the system itself.
* This task is distinct from other options because it specifically targets the transformation logic (XSLT) rather than broader integration components or services.
Let's examine why the other options are incorrect:
* A. Edit Integration Attachment: This task is used to manage generic attachments associated with an integration, such as input files or supplementary documents, but it does not specifically address XSLT transformations. It lacks the precision required for updating transformation logic.
* B. Edit Integration Attachment Service: This is not a recognized task in Workday's integration framework. It appears to be a conflation of terms and does not align with the documented processes for managing XSLT files.
* D. Edit Integration Service Attachment: While this might suggest modifying an attachment related to an integration service, it is not the correct task for handling XSLT files in a document transformation context. Workday documentation consistently points to "Edit XSLT Attachment Transformation" for this purpose.
The process typically involves:
* Navigating to the integration system in Workday (e.g., via the "Search" bar by entering the integration system name).
* Using the related actions menu to select "Integration System" > "Edit XSLT Attachment Transformation."
* Uploading the new XSLT file, which must comply with Workday's size limitations (e.g., 30 MB for attachments) and be properly formatted.
* Saving the changes, which updates the transformation logic without altering other integration configurations.
This approach ensures that transformations remain aligned with business requirements, such as reformatting data for compatibility with external systems, while leveraging Workday's secure and efficient integration tools.
References:
* Workday Pro Integrations Study Guide: "Configure Integration System - TRANSFORMATION" section, which details the use of XSLT files in document transformations and the associated tasks.
* Workday Documentation: "Enterprise Interface Builder (EIB)" and "Document Transformation Connector" sections, where the "Edit XSLT Attachment Transformation" task is outlined for updating XSLT files.
* Workday Community: Guidance on managing XSLT attachments, confirming this task as the standard method for updating pre-existing transformations.

質問 # 52
......

Workday-Pro-Integrations問題例

- Workday-Pro-Integrations問題例 □ Workday-Pro-Integrations受験記対策 □ Workday-Pro-Integrations認証資格 ↩ Open Webサイト[www.goshiken.com]検索□ Workday-Pro-Integrations □無料ダウンロードWorkday-Pro-Integrations最新試験
- Workday-Pro-Integrations試験関連情報はWorkday Pro Integrations Certification Examに合格するのが最も鋭い剣になります □今すぐ➠ www.it-passports.com □で□ Workday-Pro-Integrations □を検索し、無料でダウンロードしてくださいWorkday-Pro-Integrations最新試験
- Workday-Pro-Integrations問題無料 □ Workday-Pro-Integrations問題無料 □ Workday-Pro-Integrations問題例 □「 Workday-Pro-Integrations 」を無料でダウンロード「 www.goshiken.com 」で検索するだけWorkday-Pro-Integrations勉強時間
- Workday-Pro-Integrations日本語対策 ➡□ Workday-Pro-Integrations絶対合格 □ Workday-Pro-Integrations受験資格 □今すぐ□ www.japancert.com □で【 Workday-Pro-Integrations 】を検索し、無料でダウンロードしてくださいWorkday-Pro-Integrations日本語版と英語版
- Workday Workday-Pro-Integrations Exam | Workday-Pro-Integrations試験関連情報 - あなたのWorkday-Pro-Integrations試験の合格的なプロバイダー □ □ Workday-Pro-Integrations □の試験問題は{ www.goshiken.com }で無料配信中Workday-Pro-Integrations関連復習問題集
- Workday-Pro-Integrations問題例 □ Workday-Pro-Integrationsクラムメディア □ Workday-Pro-Integrations最新試験 □「 www.shikenpass.com 」に移動し、➠ Workday-Pro-Integrations □を検索して、無料でダウンロード可能な試験資料を探しますWorkday-Pro-Integrations日本語対策
- Workday-Pro-Integrations試験関連情報はWorkday Pro Integrations Certification Examに合格するのが最も鋭い剣になります □今すぐ▶ www.goshiken.com◀を開き、□ Workday-Pro-Integrations □を検索して無料でダウンロードしてくださいWorkday-Pro-Integrationsトレーニング
- 便利なWorkday-Pro-Integrations試験関連情報と素晴らしいWorkday-Pro-Integrations受験体験 □今すぐ[www.mogiexam.com]で➽ Workday-Pro-Integrations □を検索し、無料でダウンロードしてくださいWorkday-Pro-Integrations勉強時間
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, internshub.co.in, daotao.wisebusiness.edu.vn, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

さらに、JPTestKing Workday-Pro-Integrationsダンプの一部が現在無料で提供されています：https://drive.google.com/open?id=1DQ7zWYUYLkYX6lwjLM1mJ_-5mgcwLPf4