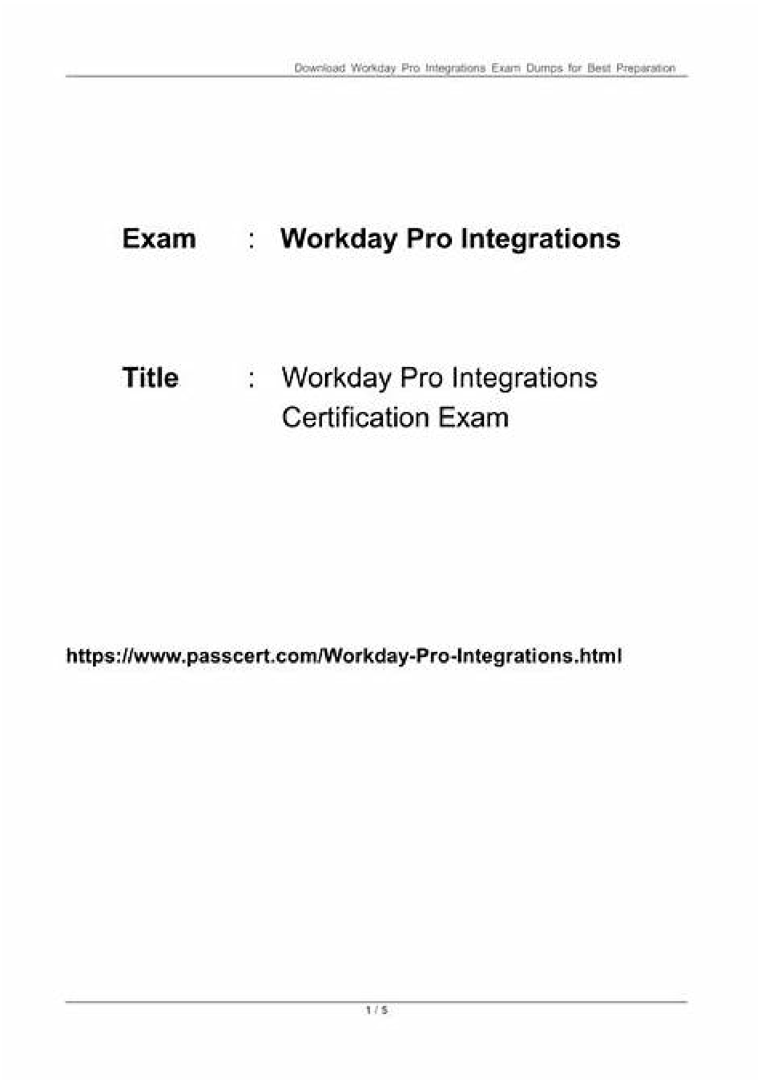
With Our Information-Packed PDF, Prepare for Workday Workday-Pro-Integrations Exam Questions



Download Workday Pro Integrations Exam Dumps for Best Preparation

Exam : Workday Pro Integrations

Title : Workday Pro Integrations Certification Exam

https://www.passcert.com/Workday-Pro-Integrations.html

1 / 5

What's more, part of that PrepAwayTest Workday-Pro-Integrations dumps now are free: https://drive.google.com/open?id=1fGXRjcczpgze8gJhc_vLtI6CETCljwNs

Workday-Pro-Integrations exam dumps provided by PrepAwayTest are tested through practice, and are the most correct and the newest practical Workday-Pro-Integrations test dumps. Our PrepAwayTest can provide accurate Workday-Pro-Integrations certification training questions based on extensive research and the experience of real world to make you pass Workday-Pro-Integrations Certification Exam in a short time. If you purchase our Workday-Pro-Integrations exam dumps, we will offer free update service within one year.

Workday Workday-Pro-Integrations Exam Syllabus Topics:

Topic	Details
Topic 1	• 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 2	• Cloud Connect: This section of the exam measures the skills of Workday Implementation Consultants and focuses on using Workday Cloud Connect solutions for third-party integration. It includes understanding pre-built connectors, configuration settings, and how to manage data flow between Workday and external systems while ensuring security and data integrity.
Topic 3	• 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 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.
Topic 5	• XSLT: This section of the exam measures the skills of Data Integration Developers and covers the use of Extensible Stylesheet Language Transformations (XSLT) in Workday integrations. It focuses on transforming XML data structures, applying conditional logic, and formatting output for various integration use cases such as APIs and external file delivery.

>> Workday-Pro-Integrations Real Question <<

Test Workday-Pro-Integrations King | Workday-Pro-Integrations Test Dumps Demo

There are different versions of our Workday-Pro-Integrations learning materials: the PDF, Software and APP online versions. Whether you like to study on the computer or like to read paper materials, our Workday-Pro-Integrationslearning materials can meet your needs. If you are used to reading paper with our Workday-Pro-Integrations Study Materials for most of the time, you can eliminate your concerns. Our Workday-Pro-Integrations exam quiz takes full account of customers' needs in this area.

Workday Pro Integrations Certification Exam Sample Questions (Q60-Q65):

NEW QUESTION # 60
What is the relationship between the Integration System User (ISU), Integration System Security Group (ISSG), and domain security policies?

- A. Assign the ISU to the ISSG, and then assign the ISSG to domain security policies.
- B. Assign domain security policies to the ISSG, and then assign the ISSG to the ISU.
- C. Assign the ISSG to the ISU, and then assign the ISU to domain security policies.
- D. Assign domain security policies to the ISU, and then assign the ISU to the ISSG.

Answer: A

Explanation:
This question is about the correct order of Workday security assignment for integrations. Workday clearly specifies the security structure:
"You assign the ISU to the Integration System Security Group (ISSG).
Then you assign the ISSG to the domain security policies."
This is because domain security policies apply to security groups, not directly to ISUs.
Correct Relationship Order:
Create ISU
Create/assign ISU to ISSG
Assign ISSG to the domain security policies (Get/Put/View)
That aligns exactly to option C.

NEW QUESTION # 61

You have a population of workers who have put multiple names in their Legal Name - First Name Workday delivered field. Your third-party vendor only accepts one-word first names. For workers that have included a middle name, the first and middle names are separated by a single space. You have been asked to implement the following logic:
* Extract the value before the single space from the Legal Name - First Name Workday delivered field.
* Count the number of characters in the extracted value.
* Identify if the number of characters is greater than.
* If the count of characters is greater than 0, use the extracted value. Otherwise, use the Legal Name - First Name Workday delivered field.
What functions are needed to achieve the end goal?

- A. Substring Text, Text Length, True/False Condition, Evaluate Expression
- B. Text Constant, Substring Text, Arithmetic Calculation, Evaluate Expression
- C. Extract Single Instance, Text Length, Numeric Constant, True/False Condition
- D. Format Text, Convert Text to Number, True/False Condition, Evaluate Expression

Answer: A

Explanation:
The task involves processing the "Legal Name - First Name" field in Workday to meet a third-party vendor's requirement of accepting only one-word first names. For workers with multiple names (e.g., "John Paul"), separated by a single space, the logic must:
* Extract the value before the space (e.g., "John" from "John Paul").
* Count the characters in the extracted value.
* Check if the character count is greater than 0.
* Use the extracted value if the count is greater than 0; otherwise, use the original "Legal Name - First Name" field.
This logic is typically implemented in Workday using calculated fields within a custom report or integration (e.g., EIB or Studio).
Let's break down the required functions:
* Substring Text:This function is needed to extract the portion of the "Legal Name - First Name" field before the space. In Workday, the Substring Text function allows you to specify a starting position (e.
g., 1) and extract text up to a delimiter (e.g., a space). For example, Substring Text("John Paul", 1, Index of " ") would return "John."
* Text Length:After extracting the substring (e.g., "John"), the logic requires counting its characters to ensure it's valid. The Text Length function returns the number of characters in a text string (e.g., Text Length("John") = 4). This is critical for the condition check.
* True/False Condition:The logic involves a conditional check: "Is the number of characters greater than
0?" The True/False Condition function evaluates this (e.g., Text Length(extracted value) > 0), returning True if the extracted value exists and False if it's empty (e.g., if no space exists or extraction fails).
* Evaluate Expression:This function implements the if-then-else logic: if the character count is greater than 0, use the extracted value (e.g., "John"); otherwise, use the original "Legal Name - First Name" field (e.g., "John Paul"). Evaluate Expression combines the True/False Condition with the output values.
* Option Analysis:
* A. Extract Single Instance, Text Length, Numeric Constant, True/False Condition:
Incorrect. Extract Single Instance is used for multi-instance fields (e.g., selecting one dependent), not text parsing. Numeric Constant isn't needed here, as no fixed number is involved.
* B. Text Constant, Substring Text, Arithmetic Calculation, Evaluate Expression: Incorrect.
Text Constant provides a fixed string (e.g., "abc"), not dynamic extraction. Arithmetic Calculation isn't required, as this is a text length check, not a numeric operation beyond comparison.
* C. Format Text, Convert Text to Number, True/False Condition, Evaluate Expression:
Incorrect. Format Text adjusts text appearance (e.g., capitalization), not extraction. Convert Text to Number isn't needed, as Text Length already returns a number.
* D. Substring Text, Text Length, True/False Condition, Evaluate Expression: Correct. These functions align perfectly with the requirements: extract the first name, count its length, check the condition, and choose the output.
* Implementation:
* Create a calculated field usingSubstring Textto extract text before the space.
* UseText Lengthto count characters in the extracted value.
* UseTrue/False Conditionto check if the length > 0.
* UseEvaluate Expressionto return the extracted value or the original field based on the condition.
References from Workday Pro Integrations Study Guide:
* Workday Calculated Fields: Section on "Text Functions" details Substring Text and Text Length usage.
* Integration System Fundamentals: Explains how calculated fields with conditions (True/False, Evaluate Expression) transform data for third-party systems.
* Core Connectors & Document Transformation: Highlights text manipulation for outbound integration requirements.

NEW QUESTION # 62
What is the purpose of the <xsl:template> element?

- A. Generate an output file name.
- B. Determine the output file type.
- C. Provide rules to apply to a specified node.
- D. Grant access to the XSLT language.

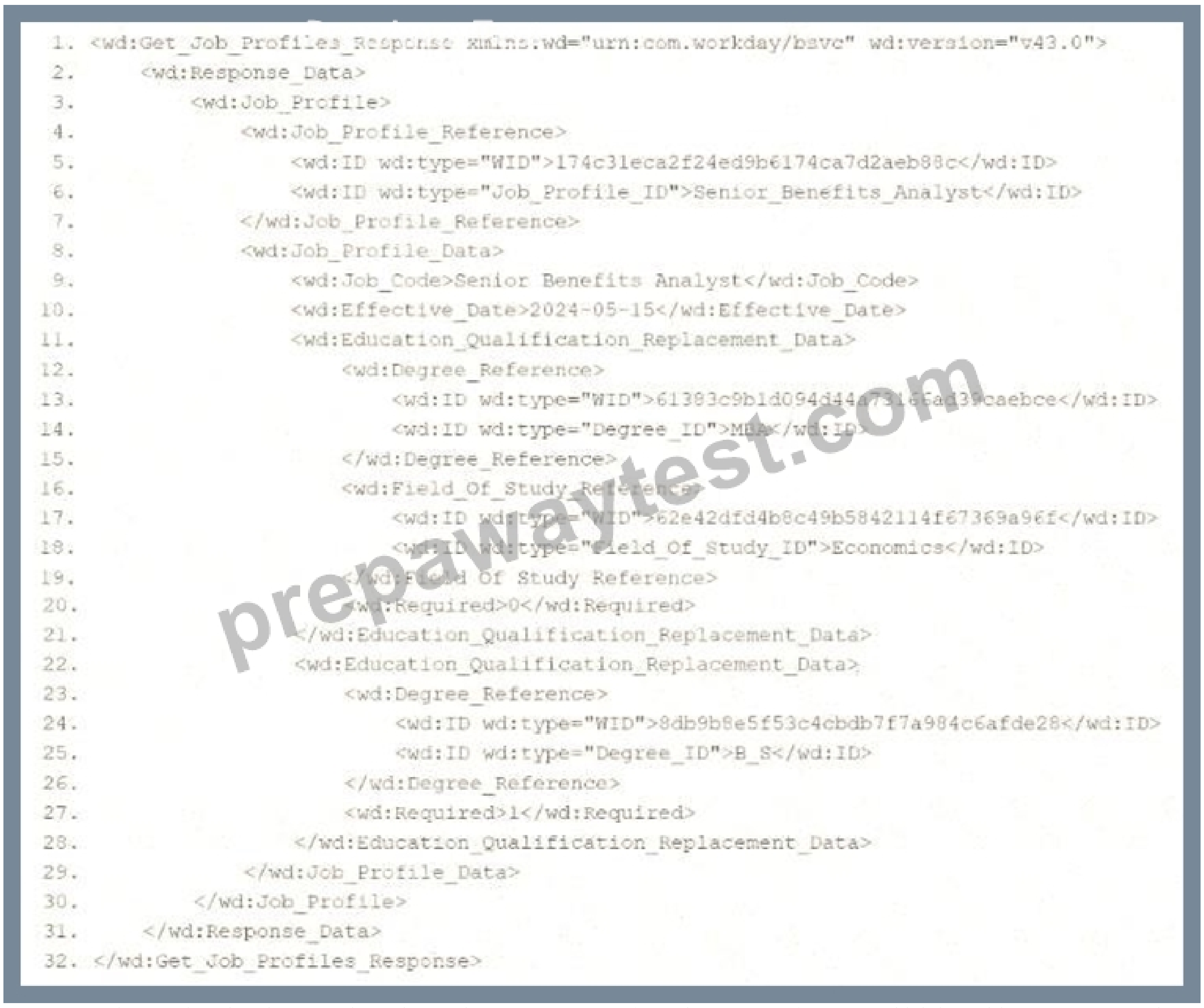
Answer: C

Explanation:
The <xsl:template> element is a fundamental component of XSLT (Extensible Stylesheet Language Transformations), which is widely used in Workday integrations, particularly within document transformation systems such as those configured via the Enterprise Interface Builder (EIB) or Document Transformation Connectors. Its primary purpose is to define rules or instructions that dictate how specific nodes in an XML source document should be processed and transformed into the desired output format.
Here's a detailed explanation of why this is the correct answer:
In XSLT, the <xsl:template> element is used to create reusable transformation rules. It typically includes a match attribute, which specifies the XML node or pattern (e.g., an element, attribute, or root node) to which the template applies. For example, <xsl:template match="Employee"> would target all <Employee> elements in the source XML.
Inside the <xsl:template> element, you define the logic-such as extracting data, restructuring it, or applying conditions-that determines how the matched node is transformed into the output. This makes it a core mechanism for controlling the transformation process in Workday integrations.
In the context of Workday, where XSLT is often used to reformat XML data into formats like CSV, JSON, or custom XML for external systems, <xsl:template> provides the structure for specifying how data from Workday's XML output (e.g., payroll or HR data) is mapped and transformed.
Let's evaluate why the other options are incorrect:
A . Determine the output file type: The <xsl:template> element does not control the output file type (e.g., XML, text, HTML). This is determined by the <xsl:output> element in the XSLT stylesheet, which defines the format of the resulting file independently of individual templates.
B . Grant access to the XSLT language: This option is nonsensical in the context of XSLT. The <xsl:template> element is part of the XSLT language itself and does not "grant access" to it; rather, it is a functional building block used within an XSLT stylesheet.
D . Generate an output file name: The <xsl:template> element has no role in naming the output file. In Workday, the output file name is typically configured within the integration system settings (e.g., via the EIB or connector configuration) and is not influenced by the XSLT transformation logic.
An example of <xsl:template> in action might look like this in a Workday transformation:

```
<xsl:template match="wd:Worker">
<Employee>
<Name><xsl:value-of select="wd:Worker_Name"/></Name>
</Employee>
</xsl:template>
```

Here, the template matches the Worker node in Workday's XML schema and transforms it into a simpler <Employee> structure with a Name element, demonstrating its role in providing rules for node transformation.
:
Workday Pro Integrations Study Guide: "Configure Integration System - TRANSFORMATION" section, which explains XSLT usage in Workday and highlights <xsl:template> as the mechanism for defining transformation rules.
Workday Documentation: "XSLT Transformations in Workday" under the Document Transformation Connector, noting <xsl:template> as critical for node-specific processing.
W3C XSLT 1.0 Specification (adopted by Workday): Section 5.3, "Defining Template Rules," which confirms that <xsl:template> provides rules for applying transformations to specified nodes.
Workday Community: Examples of XSLT in integration scenarios, consistently using <xsl:template> for transformation logic.

NEW QUESTION # 63
Refer to the following XML to answer the question below.



```
<wd:Get_Job_Profiles_Response xmlns:wd="urn:com.workday/bsvc" wd:version="v43.0">
    <wd:Response_Data>
        <wd:Job_Profile>
            <wd:Job_Profile_Reference>
                <wd:ID wd:type="WID">174c31eca2f24ed9b6174ca7d2aeb88c</wd:ID>
                <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
            </wd:Job_Profile_Reference>
            <wd:Job_Profile_Data>
                <wd:Job_Code>Senior Benefits Analyst</wd:Job_Code>
                <wd:Effective_Date>2024-05-15</wd:Effective_Date>
                <wd:Education_Qualification_Replacement_Data>
                    <wd:Degree_Reference>
                        <wd:ID wd:type="WID">61383c9b1d094d44a7316fad39caebce</wd:ID>
                        <wd:ID wd:type="Degree_ID">MBA</wd:ID>
                    </wd:Degree_Reference>
                    <wd:Field_Of_Study_Reference>
                        <wd:ID wd:type="WID">62e42dfd4b8c49b5842114f67369a96f</wd:ID>
                        <wd:ID wd:type="Field_Of_Study_ID">Economics</wd:ID>
                    </wd:Field Of Study Reference>
                    <wd:Required>0</wd:Required>
                </wd:Education_Qualification_Replacement_Data>
                <wd:Education_Qualification_Replacement_Data>
                    <wd:Degree_Reference>
                        <wd:ID wd:type="WID">8db9b8e5f53c4cbdb7f7a984c6afde28</wd:ID>
                        <wd:ID wd:type="Degree_ID">B_S</wd:ID>
                    </wd:Degree_Reference>
                    <wd:Required>1</wd:Required>
                </wd:Education_Qualification_Replacement_Data>
            </wd:Job_Profile_Data>
        </wd:Job_Profile>
    </wd:Response_Data>
</wd:Get_Job_Profiles_Response>
```

prepawaytest.com

You are an integration developer and need to write XSLT to transform the output of an EIB which is making a request to the Get Job Profiles web service operation. The root template of your XSLT matches on the <wd:
Get_Job_Profiles_Response> element. This root template then applies templates against <wd:Job_Profile>.
What XPath syntax would be used to select the value of the ID element which has a wd:type attribute named Job_Profile_ID when the <xsl:value-of> element is placed within the template which matches on <wd:
Job_Profile>?

- A. wd:Job_Profile_Reference/wd:ID/wd:type='Job_Profile_ID'
- B. wd:Job_Profile_Reference/wd:ID/@wd:type='Job_Profile_ID'
- C. wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']
- D. wd:Job_Profile_Reference/wd:ID/[@wd:type='Job_Profile_ID']

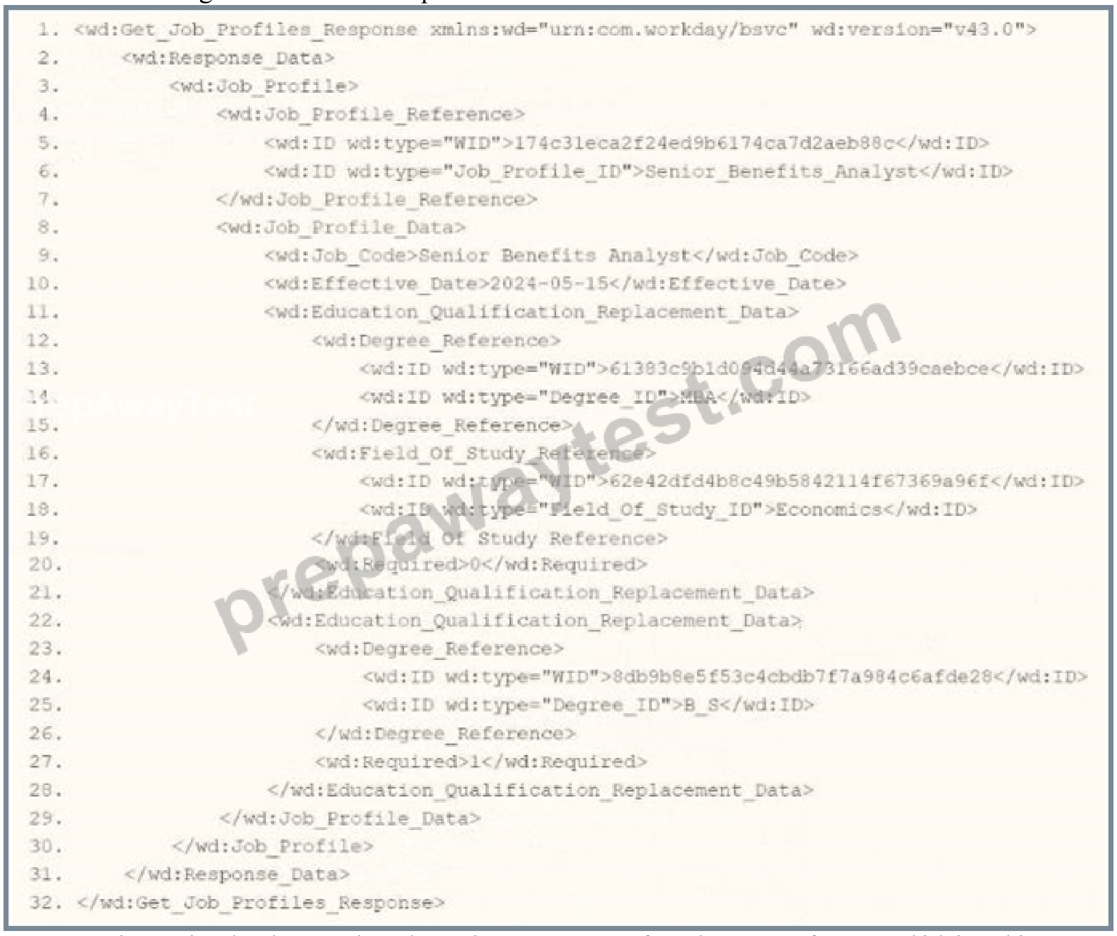
Answer: C

Explanation:
As an integration developer working with Workday, you are tasked with transforming the output of an Enterprise Interface Builder (EIB) that calls the Get_Job_Profiles web service operation. The provided XML shows the response from this operation, and you need to write XSLT to select the value of the <wd:ID> element where the wd:type attribute equals "Job_Profile_ID." The root template of your XSLT matches on
<wd:Get_Job_Profiles_Response> and applies templates to <wd:Job_Profile>. Within this template, you use the <xsl:value-of> element to extract the value. Let's analyze the XML structure, the requirement, and each option to determine the correct XPath syntax.
Understanding the XML and Requirement
The XML snippet provided is a SOAP response from the Get_Job_Profiles web service operation in Workday, using the namespace xmlns:wd="urn:com.workday/bsvc" and version wd:version="v43.0". Key elements relevant to the question include:
* The root element is <wd:Get_Job_Profiles_Response>.
* It contains <wd:Response_Data>, which includes <wd:Job_Profile> elements.
* Within <wd:Job_Profile>, there is <wd:Job_Profile_Reference>, which contains multiple <wd:ID> elements, each with a wd:type attribute:
* <wd:ID wd:type="WID">1740d3eca2f2ed9b6174ca7d2ae88c8c</wd:ID>

* <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
The task is to select the value of the <wd:ID> element where wd:type="Job_Profile_ID" (e.g., "Senior_Benefits_Analyst") using XPath within an XSLT template that matches <wd:Job_Profile>. The <xsl:value-of> element outputs the value of the selected node, so you need the correct XPath path from the <wd:Job_Profile> context to the specific <wd:ID> element with the wd:type attribute value "Job_Profile_ID." Analysis of Options Let's evaluate each option based on the XML structure and XPath syntax rules:
* Option A: wd:Job_Profile_Reference/wd:ID/wd:type='Job_Profile_ID'
* This XPath attempts to navigate from wd:Job_Profile_Reference to wd:ID, then to wd:type='Job_Profile_ID'. However, there are several issues:
* wd:type='Job_Profile_ID' is not valid XPath syntax. In XPath, to filter based on an attribute value, you use the attribute selector [@attribute='value'], not a direct comparison like wd:type='Job_Profile_ID'.
* wd:type is an attribute of <wd:ID>, not a child element or node. This syntax would not select the <wd:ID> element itself but would be interpreted as trying to match a nonexistent child node or property, resulting in an error or no match.
* This option is incorrect because it misuses XPath syntax for attribute filtering.
* Option B: wd:Job_Profile_Reference/wd:ID/@wd:type='Job_Profile_ID'
* This XPath navigates to wd:Job_Profile_Reference/wd:ID and then selects the @wd:type attribute, comparing it to "Job_Profile_ID" with =@wd:type='Job_Profile_ID'. However:
* The =@wd:type='Job_Profile_ID' syntax is invalid in XPath. To filter based on an attribute value, you use [@wd:type='Job_Profile_ID'] as a predicate, not an equality comparison in this form.
* This XPath would select the wd:type attribute itself (e.g., the string "Job_Profile_ID"), not the value of the <wd:ID> element. Since <xsl:value-of> expects a node or element value, selecting an attribute directly would not yield the desired "Senior_Benefits_Analyst" value.
* This option is incorrect due to the invalid syntax and inappropriate selection of the attribute instead of the element value.
* Option C: wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']
* This XPath navigates from wd:Job_Profile_Reference to wd:ID and uses the predicate [@wd:type='Job_Profile_ID'] to filter for <wd:ID> elements where the wd:type attribute equals "Job_Profile_ID."
* In the XML, <wd:Job_Profile_Reference> contains:
* <wd:ID wd:type="WID">1740d3eca2f2ed9b6174ca7d2ae88c8c</wd:ID>
* <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
* The predicate [@wd:type='Job_Profile_ID'] selects the second <wd:ID> element, whose value is "Senior_Benefits_Analyst."
* Since the template matches <wd:Job_Profile>, and <wd:Job_Profile_Reference> is a direct child of <wd:Job_Profile>, this path is correct:
* <wd:Job_Profile> # <wd:Job_Profile_Reference> # <wd:ID[@wd:type='Job_Profile_ID']>.
* When used with <xsl:value-of select="wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']"/>, it outputs "Senior_Benefits_Analyst," fulfilling the requirement.
* This option is correct because it uses proper XPath syntax for attribute-based filtering and selects the desired <wd:ID> value.
* Option D: wd:Job_Profile_Reference/wd:ID/[@wd:type='Job_Profile_ID']
* This XPath is similar to Option C but includes an extra forward slash before the predicate: wd:ID/[@wd:type='Job_Profile_ID']. In XPath, predicates like [@attribute='value'] are used directly after the node name (e.g., wd:ID[@wd:type='Job_Profile_ID']), not separated by a slash. The extra slash is syntactically incorrect and would result in an error or no match, as it implies navigating to a child node that doesn't exist.
* This option is incorrect due to the invalid syntax.
Why Option C is Correct
Option C, wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID'], is the correct XPath syntax because:
* It starts from the context node <wd:Job_Profile> (as the template matches this element) and navigates to <wd:Job_Profile_Reference/wd:ID>, using the predicate [@wd:type='Job_Profile_ID'] to filter for the <wd:ID> element with wd:type="Job_Profile_ID".
* It correctly selects the value "Senior_Benefits_Analyst," which is the content of the <wd:ID> element where wd:type="Job_Profile_ID".
* It uses standard XPath syntax for attribute-based filtering, aligning with Workday's XSLT implementation for web service responses.
* When used with <xsl:value-of>, it outputs the required value, fulfilling the question's requirement.
Practical Example in XSLT
Here's how this might look in your XSLT:
```
<xsl:template match="wd:Job_Profile">
<xsl:value-of select="wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']"/>
</xsl:template>
```
This would output "Senior_Benefits_Analyst" for the <wd:ID> element with wd:type="Job_Profile_ID" in the XML.

Verification with Workday Documentation
The Workday Pro Integrations Study Guide and SOAP API Reference (available via Workday Community) detail the structure of the Get_Job_Profiles response and how to use XPath in XSLT for transformations. The XML structure shows <wd:Job_Profile_Reference> containing <wd:ID> elements with wd:type attributes, and the guide emphasizes using predicates like [@wd:type='value'] to filter based on attributes. This is a standard practice for navigating Workday web service responses.
Workday Pro Integrations Study Guide References
* Section: XSLT Transformations in EIBs - Describes using XSLT to transform web service responses, including selecting elements with XPath and attribute predicates.
* Section: Workday Web Services - Details the Get_Job_Profiles operation and its XML output structure, including <wd:Job_Profile_Reference> and <wd:ID> with wd:type attributes.
* Section: XPath Syntax - Explains how to use predicates like [@wd:type='Job_Profile_ID'] for attribute- based filtering in Workday XSLT.
* Workday Community SOAP API Reference - Provides examples of XPath navigation for Workday web service responses, including attribute selection.
Option C is the verified answer, as it correctly selects the <wd:ID> value with wd:type="Job_Profile_ID" using the appropriate XPath syntax within the <wd:Job_Profile> template context.

NEW QUESTION # 64
Refer to the following XML to answer the question below.
Refer to the following XML to answer the question below.



```
<wd:Get_Job_Profiles_Response xmlns:wd="urn:com.workday/bsvc" wd:version="v43.0">
    <wd:Response_Data>
        <wd:Job_Profile>
            <wd:Job_Profile_Reference>
                <wd:ID wd:type="WID">174c31eca2f24ed9b6174ca7d2aeb88c</wd:ID>
                <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
            </wd:Job_Profile_Reference>
            <wd:Job_Profile_Data>
                <wd:Job_Code>Senior Benefits Analyst</wd:Job_Code>
                <wd:Effective_Date>2024-05-15</wd:Effective_Date>
                <wd:Education_Qualification_Replacement_Data>
                    <wd:Degree_Reference>
                        <wd:ID wd:type="WID">61383c9b1d094d44a73166ad39caebce</wd:ID>
                        <wd:ID wd:type="Degree_ID">MBA</wd:ID>
                    </wd:Degree_Reference>
                    <wd:Field_Of_Study_Reference>
                        <wd:ID wd:type="WID">62e42dfd4b8c49b5842114f67369a96f</wd:ID>
                        <wd:ID wd:type="Field_Of_Study_ID">Economics</wd:ID>
                    </wd:Field Of Study Reference>
                    <wd:Required>0</wd:Required>
                </wd:Education_Qualification_Replacement_Data>
                <wd:Education_Qualification_Replacement_Data>
                    <wd:Degree_Reference>
                        <wd:ID wd:type="WID">8db9b8e5f53c4cbdb7f7a984c6afde28</wd:ID>
                        <wd:ID wd:type="Degree_ID">B_S</wd:ID>
                    </wd:Degree_Reference>
                    <wd:Required>1</wd:Required>
                </wd:Education_Qualification_Replacement_Data>
            </wd:Job_Profile_Data>
        </wd:Job_Profile>
    </wd:Response_Data>
</wd:Get_Job_Profiles_Response>
```

prepawaytest.com

You are an integration developer and need to write XSLT to transform the output of an EIB which is making a request to the Get Job Profiles web service operation. The root template of your XSLT matches on the <wd:Get_Job_Profiles_Response> element. This root template then applies templates against <wd:Job_Profile>. XPath contains a number of delivered functions such as format-date. The format-date function uses the following syntax: format-date ($value as xs: date? $picture as xs:string). Within the template which matches on <wd:Job_Profile>, what XPath syntax would you use to output the value of the <wd:Effective_Date> element formatted with the day-month-year format of "15-07-2024"?

- A. format-date (wd:Job_Profile_Data/wd:Effective_Date, '[D01]-[M01]-[Y0001]')
- B. format-date (wd:Job_Profile_Data/wd:Effective_Date, '[M01]-[D01]-[Y0001]')
- C. format-date('[D01]-[M01]-[Y0001]', wd:Job_Profile_Data/wd:Effective_Date)

- D. format-date('[M01]-[D01]-[Y0001]', wd:Job_Profile_Data/wd:Effective_Date)

Answer: A

Explanation:
As an integration developer working with Workday, you are tasked with transforming the output of an Enterprise Interface Builder (EIB) that calls the Get_Job_Profiles web service operation. The XML provided shows the response from this operation, and you need to write XSLT to format the <wd:Effective_Date> element within the <wd:Job_Profile_Data> section. Specifically, you need to output the date "2024-05-15" (as seen in the XML) in the format "15-07-2024" (day-month-year). The root template of your XSLT matches on <wd:Get_Job_Profiles_Response> and applies templates to <wd:Job_Profile>. You are using the format-date XPath function, which follows the syntax: format-date($value as xs:date?, $picture as xs:string). Let's analyze the XML, the requirement, and each option to determine the correct XPath syntax.
Understanding the XML and Requirement
The provided XML snippet shows a response from the Get_Job_Profiles web service operation in Workday, formatted in SOAP XML with the Workday namespace (xmlns:wd="urn:com.workday/bsvc"). Key elements relevant to the question include:
The root element is <wd:Get_Job_Profiles_Response>.
It contains <wd:Response_Data>, which includes <wd:Job_Profile> elements.
Within <wd:Job_Profile>, there is <wd:Job_Profile_Data>, which contains <wd:Effective_Date> with the value 2024-05-15.
You need to transform this date into the format "15-07-2024" (DD-MM-YYYY), where:
"15" is the day (D01 for two digits).
"07" is the month (M01 for two digits, noting the XML shows May, but the question specifies July for the output format-likely a hypothetical or test case adjustment).
"2024" is the year (Y0001 for four digits).
The format-date function in XPath 2.0 (used by Workday) formats a date value according to a picture string. The syntax is:
First parameter: The date value (e.g., wd:Job_Profile_Data/wd:Effective_Date), which must be an xs:date or convertible to one.
Second parameter: The picture string (e.g., '[D01]-[M01]-[Y0001]'), specifying the format using patterns like:
[D01] for two-digit day (01-31).
[M01] for two-digit month (01-12).
[Y0001] for four-digit year (e.g., 2024).
The question specifies that the root template matches <wd:Get_Job_Profiles_Response> and applies templates to <wd:Job_Profile>, so the XPath must navigate to <wd:Job_Profile_Data/wd:Effective_Date> within that context.
Analysis of Options
Let's evaluate each option based on the format-date syntax, the XML structure, and the required output format "15-07-2024":
Option A: format-date('[D01]-[M01]-[Y0001]', wd:Job_Profile_Data/wd:Effective_Date) This option places the picture string ('[D01]-[M01]-[Y0001]') as the first parameter and the date value (wd:Job_Profile_Data/wd:Effective_Date) as the second. However, the format-date function requires the date value as the first parameter and the picture string as the second, per the syntax format-date($value, $picture). Reversing the parameters is incorrect and will result in an error or unexpected output, as format-date expects an xs:date? first. Thus, this option is invalid.
Option B: format-date (wd:Job_Profile_Data/wd:Effective_Date, '[D01]-[M01]-[Y0001]') This option correctly follows the format-date syntax:
First parameter: wd:Job_Profile_Data/wd:Effective_Date, which points to the <wd:Effective_Date> element in the XML (e.g., 2024-05-15). This is an xs:date value, as Workday web services typically return dates in ISO format (YYYY-MM-DD), which format-date can process.
Second parameter: '[D01]-[M01]-[Y0001]', which specifies the output format:
[D01] outputs the day as two digits (e.g., "15").
[M01] outputs the month as two digits (e.g., "05" for May, but the question requests "07" for July-assuming a test case adjustment or hypothetical transformation).
[Y0001] outputs the year as four digits (e.g., "2024").
The XPath wd:Job_Profile_Data/wd:Effective_Date is correctly nested under the <wd:Job_Profile> context, as the template matches on <wd:Job_Profile>. This would transform "2024-05-15" into "15-05-2024" (or "15-07-2024" if the month is adjusted in the logic), matching the required day-month-year format. This option is valid and correct.
Option C: format-date (wd:Job_Profile_Data/wd:Effective_Date, '[M01]-[D01]-[Y0001]') This option also follows the correct format-date syntax, with the date value first and the picture string second. However, the picture string '[M01]-[D01]-[Y0001]' specifies a month-day-year format:
[M01] outputs the month first (e.g., "05" for May).
[D01] outputs the day second (e.g., "15").
[Y0001] outputs the year last (e.g., "2024").
This would transform "2024-05-15" into "05-15-2024," which does not match the required "15-07-2024" (day-month-year) format. Thus, this option is incorrect for the specified output.
Option D: format-date('[M01]-[D01]-[Y0001]', wd:Job_Profile_Data/wd:Effective_Date) Similar to Option A, this option reverses the parameters, placing the picture string ('[M01]-[D01]-[Y0001]') first and the date value

(wd:Job_Profile_Data/wd:Effective_Date) second. As explained earlier, format-date requires the date value as the first parameter, so this syntax is incorrect and will not work as intended. This option is invalid.
Why Option B is Correct
Option B correctly uses the format-date function with the proper syntax:
It places the date value (wd:Job_Profile_Data/wd:Effective_Date) as the first parameter, referencing the <wd:Effective_Date> element in the XML.
It uses the picture string '[D01]-[M01]-[Y0001]' as the second parameter, which formats the date as "DD-MM-YYYY" (e.g., "15-05-2024" for the XML's "2024-05-15," or "15-07-2024" as specified, assuming a month adjustment in the transformation logic).
The XPath is appropriate for the context, as the template matches <wd:Job_Profile>, and <wd:Job_Profile_Data/wd:Effective_Date> is a valid path within it.
The question's mention of "15-07-2024" suggests either a hypothetical adjustment (e.g., the EIB or XSLT logic modifies the month to July) or a test case variation. Since the XML shows "2024-05-15," the format-date function would output "15-05-2024" with the given picture string, but the principle of formatting day-month-year remains correct. Workday's XSLT implementation supports such transformations, and the format-date function is well-documented for this purpose.
Practical Example in XSLT
Here's how this might look in your XSLT:
<xsl:template match="wd:Job_Profile">
<xsl:value-of select="format-date(wd:Job_Profile_Data/wd:Effective_Date, '[D01]-[M01]-[Y0001]')"/>
</xsl:template>
This would process the <wd:Effective_Date> (e.g., "2024-05-15") and output "15-05-2024," aligning with the day-month-year format requested (adjusted for the hypothetical "07" if needed elsewhere in the logic).
Verification with Workday Documentation
The Workday Pro Integrations Study Guide and SOAP API Reference (available via Workday Community) detail the use of XPath functions like format-date for transforming web service responses. The Get_Job_Profiles operation returns job profile data, including effective dates, in ISO format, and XSLT transformations are commonly used in EIBs to reformat data. The format-date function's syntax and picture string patterns (e.g., [D01], [M01], [Y0001]) are standard in XPath 2.0, as implemented in Workday's integration tools.
Workday Pro Integrations Study Guide Reference
Section: XSLT Transformations in EIBs - Describes using XSLT to transform web service responses, including date formatting with format-date.
Section: Workday Web Services - Details the Get_Job_Profiles operation and its XML output structure, including <wd:Effective_Date>.
Section: XPath Functions - Explains the syntax and usage of format-date($value, $picture), including picture string patterns like [D01], [M01], and [Y0001].
Workday Community SOAP API Reference - Provides examples of date formatting in XSLT for Workday web services.
Option B is the verified answer, as it correctly applies the format-date function to format the <wd:Effective_Date> in the required day-month-year format.

NEW QUESTION # 65
......

PrepAwayTest is an excellent source of information on IT Certifications. In the PrepAwayTest, you can find study skills and learning materials for your exam. PrepAwayTest's Workday Workday-Pro-Integrations training materials are studied by the experienced IT experts. It has a strong accuracy and logic. To encounter PrepAwayTest, you will encounter the best training materials. You can rest assured that using our Workday Workday-Pro-Integrations Exam Training materials. With it, you have done fully prepared to meet this exam.

Test Workday-Pro-Integrations King: https://www.prepawaytest.com/Workday/Workday-Pro-Integrations-practice-exam-dumps.html

- Vce Workday-Pro-Integrations Free 🔲 Workday-Pro-Integrations Valid Study Guide 🔲 Workday-Pro-Integrations Test Torrent 🔲 Search on ☀ www.vce4dumps.com 🔲☀🔲 for [Workday-Pro-Integrations] to obtain exam materials for free download 🔲Workday-Pro-Integrations Authorized Test Dumps
- Workday-Pro-Integrations Reliable Exam Papers 🔲 New Workday-Pro-Integrations Dumps Ppt 🔲 Vce Workday-Pro-Integrations Free 🔲 Search on ✔ www.pdfvce.com 🔲✔🔲 for { Workday-Pro-Integrations } to obtain exam materials for free download 🔲Workday-Pro-Integrations Reliable Exam Papers
- Reliable Workday-Pro-Integrations Study Guide 🔲 Workday-Pro-Integrations Authorized Test Dumps ✈ Workday-Pro-Integrations New Dumps Ebook 🔲 Download 【 Workday-Pro-Integrations 】 for free by simply entering 🔲 www.dumpsquestion.com 🔲 website 🔲Workday-Pro-Integrations Valid Test Syllabus
- Quiz 2026 Workday Fantastic Workday-Pro-Integrations Real Question 🔲 Download " Workday-Pro-Integrations " for

free by simply searching on ➠ www.pdfvce.com □ □Workday-Pro-Integrations Test Torrent

- 2026 Updated Workday Workday-Pro-Integrations: Workday Pro Integrations Certification Exam Real Question □ Download ▹ Workday-Pro-Integrations ◃ for free by simply searching on { www.examcollectionpass.com } □Vce Workday-Pro-Integrations Free
- Workday-Pro-Integrations Latest Dumps Pdf □ Vce Workday-Pro-Integrations Free □ Workday-Pro-Integrations Reliable Exam Papers □ Search for ➠ Workday-Pro-Integrations □ and download it for free immediately on □ www.pdfvce.com □ □Workday-Pro-Integrations New Dumps Ebook
- Workday-Pro-Integrations Valid Test Syllabus □ Workday-Pro-Integrations Reliable Test Bootcamp □ Valid Workday-Pro-Integrations Test Registration □ Download 【 Workday-Pro-Integrations 】 for free by simply entering { www.exam4labs.com } website □Exam Dumps Workday-Pro-Integrations Provider
- 100% Pass Workday - Updated Workday-Pro-Integrations - Workday Pro Integrations Certification Exam Real Question □ Search for （ Workday-Pro-Integrations ） and obtain a free download on ▶ www.pdfvce.com ◀ □Workday-Pro-Integrations Latest Dumps Pdf
- Valid Workday-Pro-Integrations Test Registration □ Valid Workday-Pro-Integrations Torrent □ Vce Workday-Pro-Integrations Free □ Search for ⇛ Workday-Pro-Integrations ⇚ and download it for free immediately on " www.prepawayexam.com " □Vce Workday-Pro-Integrations Free
- Quiz 2026 Workday Fantastic Workday-Pro-Integrations Real Question □ Open " www.pdfvce.com " enter □ Workday-Pro-Integrations □ and obtain a free download □Workday-Pro-Integrations Reliable Test Bootcamp
- Latest Workday-Pro-Integrations Test Vce □ Workday-Pro-Integrations Accurate Study Material □ Workday-Pro-Integrations Test Torrent □ Immediately open ⇛ www.troytecdumps.com ⇚ and search for 「 Workday-Pro-Integrations 」 to obtain a free download □Valid Workday-Pro-Integrations Torrent
- tutor1.gerta.pl, www.courses.clinthiggs.com, 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.pcsq28.com, justpaste.me, 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, learn.csisafety.com.au, nualkale.blogspot.com, bbs.t-firefly.com, Disposable vapes

What's more, part of that PrepAwayTest Workday-Pro-Integrations dumps now are free: https://drive.google.com/open?id=1fGXRjcczpgze8gJhc_vLtI6CETCljwNs