

Free PDF Quiz 2026 Workday-Pro-Integrations: Workday Pro Integrations Certification Exam Newest Test Vce



Workday Workday-Pro-Integrations Workday Pro Integrations Certification Exam

For More Information – Visit link below:

<https://www.examsempire.com/>

Product Version

1. Up to Date products, reliable and verified.
2. Questions and Answers in PDF Format.



<https://examsempire.com/>

Visit us at: <https://www.examsempire.com/workday-pro-integrations>

2026 Latest PassLeaderVCE Workday-Pro-Integrations PDF Dumps and Workday-Pro-Integrations Exam Engine Free Share:
https://drive.google.com/open?id=1TmKxOEzgeX3nwhbyXnmwewcwT_VOYnm2

Generally speaking, the clients will pass the test if they have finished learning our Workday-Pro-Integrations test guide with no doubts. The odds to fail in the test are approximate to zero. But to guarantee that our clients won't suffer the loss we will refund the clients at once if they fail in the test unexpectedly. The procedures are very simple and the clients only need to send us their proofs to fail in the Workday-Pro-Integrations test and the screenshot or the scanning copies of the clients' failure scores. The clients can consult our online customer staff about how to refund, when will the money be returned backed to them and if they can get the full refund or they can send us mails to consult these issues.

Workday Workday-Pro-Integrations Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none"> • 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 2	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 4	<ul style="list-style-type: none"> • 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.
Topic 5	<ul style="list-style-type: none"> • 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.

>> **Workday-Pro-Integrations Test Vce** <<

New Workday-Pro-Integrations Test Simulator & Valid Test Workday-Pro-Integrations Bootcamp

The user-friendly interface of Workday-Pro-Integrations Dumps (desktop & web-based) will make your preparation effective. The PassLeaderVCE ensures that the Workday-Pro-Integrations practice exam will make you competent enough to crack the in-demand Workday-Pro-Integrations examination on the first attempt. Real Workday Workday-Pro-Integrations dumps of PassLeaderVCE come in PDF format as well.

Workday Pro Integrations Certification Exam Sample Questions (Q42-Q47):

NEW QUESTION # 42

Refer to the scenario. You are configuring a Core Connector: Worker integration to extract worker demographic and contact information. The integration uses the Data Initialization Service (DIS) and must include worker fields such as name, address, and a calculated field identifying workers eligible for a phone allowance.

During a Full File test run, the output file is missing all address-related information, even though the Address Line Data, Municipality, Region, and Postal Code fields were configured in the Configure Integration Field Attributes step. You also confirmed that the Worker Personal Data Section is marked as Include in Output.

What should you do to resolve this issue?

- A. Enable the Worker Personal Data Section Fields integration service within the Configure Integration Services step.
- B. Within the Configure Integration Services task, select the Enable All Services checkbox.
- **C. Enable the Address Data subfolder in Configure Integration Field Attributes and then reselect the address fields.**
- D. Mark each address field in the Address Data subfolder as Required in Configure Integration Field Attributes.

Answer: C

Explanation:

This question concerns a Full File test of a Core Connector: Worker integration where address fields (Address Line, Municipality, Region, Postal Code) are missing from the output, despite being configured in Configure Integration Field Attributes. Additionally, the Worker Personal Data Section is marked as Include in Output.

This issue commonly stems from a missed Enablement of the Address Data subfolder, which acts as a container for the address-related fields. Even if individual fields are selected, they will not appear in the output if their parent subfolder is not enabled.

From the Workday Pro Integrations documentation:

"Each subfolder in the integration field hierarchy, such as Address Data under Worker Personal Data, must be explicitly enabled. If the subfolder itself is not enabled, the fields within it, even if marked as Required or Included, will not be rendered in the output." To

resolve this:

Navigate to Configure Integration Field Attributes

Expand the Worker Personal Data > Address Data subfolder

Enable the subfolder

Then reselect the required address fields

Incorrect Options Explained:

A . Mark each address field as RequiredMarking fields as Required is only effective if the parent subfolder is enabled. Without enabling the subfolder, fields remain excluded.

C . Enable the Worker Personal Data Section Fields integration serviceThis pertains to service execution, not field visibility. The issue lies in field hierarchy and inclusion, not the service configuration.

D . Enable All Services in Configure Integration ServicesThis enables all integration services but does not impact field inclusion or subfolder visibility within field attribute configuration.

Reference:

Workday Pro: Integrations - Field Attributes Configuration and Subfolder Enablement Workday Community: Integration Field Attributes - Common Issues with Address Data Core Connector Deployment Guide - Field Selection and Troubleshooting

NEW QUESTION # 43

What is the purpose of a namespace in the context of a stylesheet?

- A. Indicates the start and end tag names to output.
- B. Restricts the data the processor can access.
- C. Provides elements you can use in your code.
- D. Controls the filename of the transformed result.

Answer: C

Explanation:

In the context of a stylesheet, particularly within Workday's Document Transformation system where XSLT (Extensible Stylesheet Language Transformations) is commonly used, a namespace serves a critical role in defining the scope and identity of elements and attributes. The correct answer, as aligned with Workday's integration practices and standard XSLT principles, is that a namespace "provides elements you can use in your code." Here's a detailed explanation:

Definition and Purpose of a Namespace:

A namespace in an XML-based stylesheet (like XSLT) is a mechanism to avoid naming conflicts by grouping elements and attributes under a unique identifier, typically a URI (Uniform Resource Identifier). This allows different vocabularies or schemas to coexist within the same document or transformation process without ambiguity.

In XSLT, namespaces are declared in the stylesheet using the `xmlns` attribute (e.g., `xmlns:xs="http://www.w3.org/1999/XSL/Transform"` for XSLT itself). These declarations define the set of elements and functions available for use in the stylesheet, such as `<xsl:template>`, `<xsl:value-of>`, or `<xsl:for-each>`.

For example, when transforming Workday data (which uses its own XML schema), a namespace might be defined to reference Workday-specific elements, enabling the stylesheet to correctly identify and manipulate those elements.

Application in Workday Context:

In Workday's Document Transformation integrations, namespaces are essential when processing XML data from Workday (e.g., Core Connector outputs) or external systems. The namespace ensures that the XSLT processor recognizes the correct elements from the source XML and applies the transformation rules appropriately.

Without a namespace, the processor might misinterpret elements with the same name but different meanings (e.g., `<name>` in one schema vs. another). By providing a namespace, the stylesheet gains access to a specific vocabulary of elements and attributes, enabling precise coding of transformation logic.

Why Other Options Are Incorrect:

B . Indicates the start and end tag names to output: This is incorrect because namespaces do not dictate the structure (start and end tags) of the output. That is determined by the XSLT template rules and output instructions (e.g., `<xsl:output>` or literal result elements). Namespaces only define the identity of elements, not their placement or formatting in the output.

C . Restricts the data the processor can access: While namespaces help distinguish between different sets of elements, they do not inherently restrict data access. Restrictions are more a function of security settings or XPath expressions within the stylesheet, not the namespace itself.

D . Controls the filename of the transformed result: Namespaces have no bearing on the filename of the output. In Workday, the filename of a transformed result is typically managed by the Integration Attachment Service or delivery settings (e.g., SFTP or email configurations), not the stylesheet's namespace.

Practical Example:

Suppose you're transforming a Workday XML file containing employee data into a custom format. The stylesheet might include:

```
<xsl:stylesheet version="1.0" xmlns:xs="http://www.w3.org/1999/XSL/Transform" xmlns:wd="http://www.workday.com/ns">
```

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

Here, the wd namespace provides access to Workday-specific elements like <wd:Employee> and <wd:Name>, which the XSLT processor can then use to extract and transform data.

Workday Pro Integrations Study Guide Reference:

Workday Integration System Fundamentals: Explains XML and XSLT basics, including the role of namespaces in identifying elements within stylesheets.

Document Transformation Module: Highlights how namespaces are used in XSLT to process Workday XML data, emphasizing their role in providing a vocabulary for transformation logic (e.g., "Understanding XSLT Namespaces").

Core Connectors and Document Transformation Course Manual: Includes examples of XSLT stylesheets where namespaces are declared to handle Workday-specific schemas, reinforcing that they provide usable elements.

Workday Community Documentation: Notes that namespaces are critical for ensuring compatibility between Workday's XML output and external system requirements in transformation scenarios.

NEW QUESTION # 44

Refer to the scenario. You are configuring a Core Connector: Worker integration with the Data Initialization Service (DIS) enabled. The integration must extract worker contact details and job information, including a calculated field override that determines phone allowance eligibility.

When testing, you run the Test Security Related Action from the Configure Integration Field Override step.

Several field overrides display "No" in the Available by User column.

To ensure the ISSG has access to these field overrides and that "Yes" is displayed in the Test Security step, what configuration should you review?

- A. Assign the ISSG to the domain security policies that govern the web service operations with Get access.
- **B. Provide the ISSG View permissions to the domain security policies securing each overridden field.**
- C. Identify the domain security policies securing the field overrides and grant Modify permissions.
- D. Grant View permissions to the ISSG for the domain security policies that secure the web service operations.

Answer: B

Explanation:

The Test Security Related Action shows Available by User = No when the security group running the integration lacks View permissions to the fields used in the override logic.

From Workday documentation:

Field Overrides require the ISSG to have View access to the domain policies securing each field referenced in the override, otherwise Workday blocks the field from execution.

Therefore, the appropriate fix is to:

- * Identify the domains that secure the calculated fields and overridden fields
- * Grant the ISSG View access in those domain security policies
- * Activate pending changes

Options B and C incorrectly focus only on web service operations.

Option D incorrectly suggests Modify access - but View is the required minimum.

References:Admin#Guide#Authentication#and#Security.pdf - Access to Workday Data: View access required for outbound integration fields

NEW QUESTION # 45

Refer to the following XML to answer the question below.

```

1. <wd:Get_Job_Profiles_Response xmlns:wd="urn:com.workday/bsvc" wd:version="v43.0">
2.   <wd:Response_Data>
3.     <wd:Job_Profile>
4.       <wd:Job_Profile_Reference>
5.         <wd:ID wd:type="WID">174c31eca2f24ed9b6174ca7d2ae88c</wd:ID>
6.         <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
7.       </wd:Job_Profile_Reference>
8.       <wd:Job_Profile_Data>
9.         <wd:Job_Code>Senior Benefits Analyst</wd:Job_Code>
10.        <wd:Effective_Date>2024-05-15</wd:Effective_Date>
11.        <wd:Education_Qualification_Replacement_Data>
12.          <wd:Degree_Reference>
13.            <wd:ID wd:type="WID">61383e9f1d094d44a73166ad39caebce</wd:ID>
14.            <wd:ID wd:type="Degree_ID">MBA</wd:ID>
15.          </wd:Degree_Reference>
16.          <wd:Field_Of_Study_Reference>
17.            <wd:ID wd:type="WID">62e42dfd4b8c49b5842114f67369a96f</wd:ID>
18.            <wd:ID wd:type="Field_Of_Study_ID">Economics</wd:ID>
19.          </wd:Field_Of_Study_Reference>
20.          <wd:Required></wd:Required>
21.        </wd:Education_Qualification_Replacement_Data>
22.        <wd:Education_Qualification_Replacement_Data>
23.          <wd:Degree_Reference>
24.            <wd:ID wd:type="WID">8db9b8e5f53c4cbdb7f7a984c6afde28</wd:ID>
25.            <wd:ID wd:type="Degree_ID">B_S</wd:ID>
26.          </wd:Degree_Reference>
27.          <wd:Required>1</wd:Required>
28.        </wd:Education_Qualification_Replacement_Data>
29.      </wd:Job_Profile_Data>
30.    </wd:Job_Profile>
31.  </wd:Response_Data>
32. </wd:Get_Job_Profiles_Response>

```

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 # 46

Refer to the following XML to answer the question below.

Refer to the following XML to answer the question below.

```

1. <wd:Get_Job_Profiles_Response xmlns:wd="urn:com.workday/bsvc" wd:version="v43.0">
2.   <wd:Response_Data>
3.     <wd:Job_Profile>
4.       <wd:Job_Profile_Reference>
5.         <wd:ID wd:type="WID">174c31eca2f24ed9b6174ca7d2aeb88c</wd:ID>
6.         <wd:ID wd:type="Job_Profile_ID">Senior_Benefits_Analyst</wd:ID>
7.       </wd:Job_Profile_Reference>
8.       <wd:Job_Profile_Data>
9.         <wd:Job_Code>Senior Benefits Analyst</wd:Job_Code>
10.        <wd:Effective_Date>2024-05-15</wd:Effective_Date>
11.        <wd:Education_Qualification_Replacement_Data>
12.          <wd:Degree_Reference>
13.            <wd:ID wd:type="WID">61383c9b1d094d44a73166ad39caebce</wd:ID>
14.            <wd:ID wd:type="Degree_ID">MBA</wd:ID>
15.          </wd:Degree_Reference>
16.          <wd:Field_Of_Study_Reference>
17.            <wd:ID wd:type="WID">62e42dfd4b8c49b5842114f67369a96f</wd:ID>
18.            <wd:ID wd:type="Field_Of_Study_ID">Economics</wd:ID>
19.          </wd:Field_Of_Study_Reference>
20.          <wd:Required>0</wd:Required>
21.        </wd:Education_Qualification_Replacement_Data>
22.        <wd:Education_Qualification_Replacement_Data>
23.          <wd:Degree_Reference>
24.            <wd:ID wd:type="WID">8db9b8e5f53c4cbdb7f7a984c6afde28</wd:ID>
25.            <wd:ID wd:type="Degree_ID">B_S</wd:ID>
26.          </wd:Degree_Reference>
27.          <wd:Required>1</wd:Required>
28.        </wd:Education_Qualification_Replacement_Data>
29.      </wd:Job_Profile_Data>
30.    </wd:Job_Profile>
31.  </wd:Response_Data>
32. </wd:Get_Job_Profiles_Response>

```

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

Answer: B

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 # 47

.....

If you choose to use the software version of Workday Workday-Pro-Integrations study guide, you will find that you can download our Workday Pro Integrations Certification Exam Workday-Pro-Integrations exam prep on more than one computer and you can practice our Workday-Pro-Integrations exam questions offline as well. We strongly believe that the software version of our Workday-Pro-Integrations Study Materials will be of great importance for you to prepare for the exam and all of the employees in our company wish you early success!

New Workday-Pro-Integrations Test Simulator: <https://www.passleadervce.com/Workday-Integrations/reliable-Workday-Pro-Integrations-exam-learning-guide.html>

- Reliable Workday-Pro-Integrations Test Objectives Valid Workday-Pro-Integrations Exam Questions Reliable Workday-Pro-Integrations Test Objectives The page for free download of Workday-Pro-Integrations on www.verifieddumps.com will open immediately Workday-Pro-Integrations Exam Dumps Collection
- Workday-Pro-Integrations PdfFormat Workday-Pro-Integrations Vce Format New Workday-Pro-Integrations Study Plan Enter www.pdfvce.com and search for [Workday-Pro-Integrations](#) to download for free Reliable Workday-Pro-Integrations Test Objectives
- Validate Your Skills with Workday Workday-Pro-Integrations Exam Dumps Simply search for [Workday-Pro-Integrations](#) for free download on www.prep4sures.top Latest Workday-Pro-Integrations Test Prep
- Workday-Pro-Integrations Reliable Exam Topics Latest Workday-Pro-Integrations Test Prep Workday-Pro-Integrations Latest Test Cram Immediately open www.pdfvce.com and search for [Workday-Pro-Integrations](#) to obtain a free download Workday-Pro-Integrations Reliable Braindumps
- New Workday-Pro-Integrations Study Plan Workday-Pro-Integrations Reliable Exam Topics Latest Workday-

Pro-Integrations Practice Questions □ Go to website □ www.prep4sures.top □ open and search for { Workday-Pro-Integrations } to download for free □ Workday-Pro-Integrations Vce Format

- 100% Pass 2026 Workday Workday-Pro-Integrations: Unparalleled Workday Pro Integrations Certification Exam Test Vce □ Search for « Workday-Pro-Integrations » on ▷ www.pdfvce.com ◁ immediately to obtain a free download □ □ Workday-Pro-Integrations Reliable Exam Topics
- Pass-Sure Workday-Pro-Integrations Test Vce, Ensure to pass the Workday-Pro-Integrations Exam □ Download [Workday-Pro-Integrations] for free by simply entering “ www.vce4dumps.com ” website □ Workday-Pro-Integrations Reliable Exam Topics
- Free PDF Workday - Useful Workday-Pro-Integrations - Workday Pro Integrations Certification Exam Test Vce □ ➡ www.pdfvce.com □ □ □ is best website to obtain □ Workday-Pro-Integrations □ for free download □ Dumps Workday-Pro-Integrations Free
- Workday-Pro-Integrations Reliable Braindumps □ Workday-Pro-Integrations Trustworthy Exam Content □ Workday-Pro-Integrations Valid Test Pattern □ Easily obtain □ Workday-Pro-Integrations □ for free download through [www.prepawaypdf.com] □ Workday-Pro-Integrations Vce Format
- Reliable Exam Workday-Pro-Integrations Pass4sure □ Latest Workday-Pro-Integrations Test Prep □ Latest Workday-Pro-Integrations Test Prep □ Enter [www.pdfvce.com] and search for [Workday-Pro-Integrations] to download for free □ Reliable Exam Workday-Pro-Integrations Pass4sure
- Workday-Pro-Integrations Trustworthy Exam Content □ Reliable Exam Workday-Pro-Integrations Pass4sure □ Valid Workday-Pro-Integrations Exam Questions □ Easily obtain free download of (Workday-Pro-Integrations) by searching on ➡ www.prepawayete.com □ □ Workday-Pro-Integrations Vce Format
- whitebookmarks.com, joshrggw387073.blogoxo.com, stevetfih906003.wikikarts.com, course.mbonisi.com, active-bookmarks.com, sachinewke512687.bloggactif.com, lewiswjch938496.blog-mall.com, royxgrk050508.snack-blog.com, bookmarkblast.com, harleydaq814248.wikilowdown.com, Disposable vapes

2026 Latest PassLeaderVCE Workday-Pro-Integrations PDF Dumps and Workday-Pro-Integrations Exam Engine Free Share:
https://drive.google.com/open?id=1TmKxOEzgeX3nwhbyXmwewcwT_VOYnm2