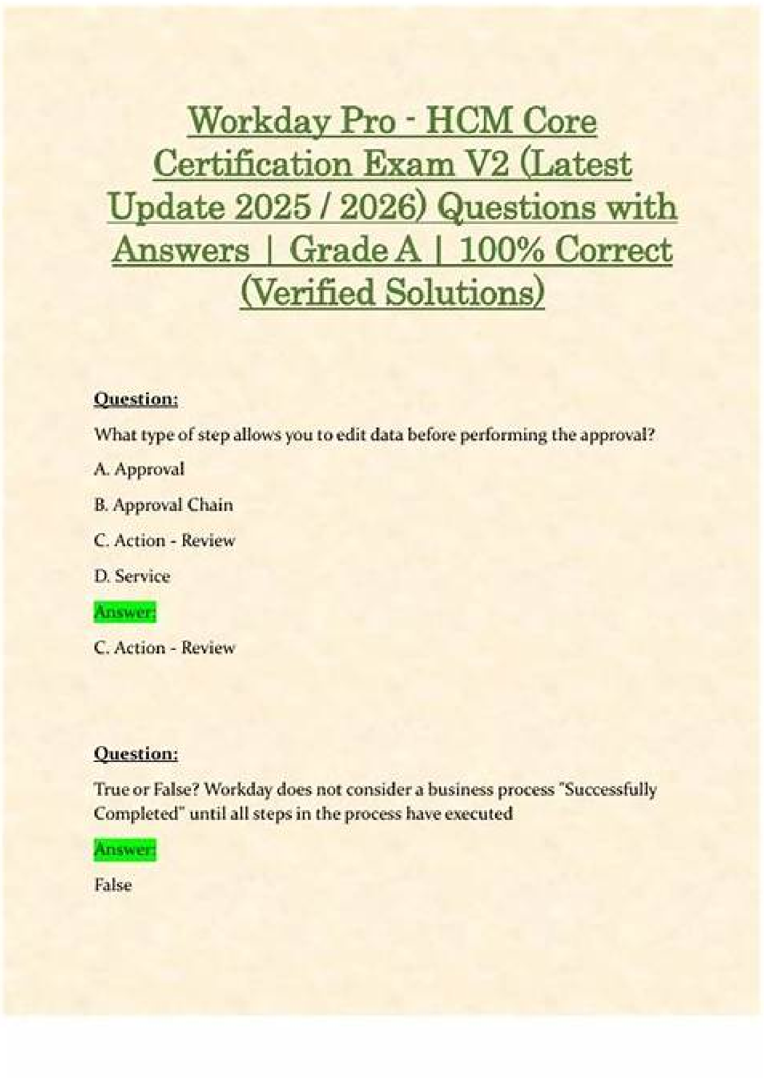
Workday certification Workday-Pro-Integrations exam questions and answers come out


Workday Pro - HCM Core Certification Exam V2 (Latest Update 2025 / 2026) Questions with Answers | Grade A | 100% Correct (Verified Solutions)

Question:

What type of step allows you to edit data before performing the approval?

A. Approval

B. Approval Chain

C. Action - Review

D. Service

Answer:

C. Action - Review

Question:

True or False? Workday does not consider a business process "Successfully Completed" until all steps in the process have executed

Answer:

False

2025 Latest PassReview Workday-Pro-Integrations PDF Dumps and Workday-Pro-Integrations Exam Engine Free Share: https://drive.google.com/open?id=1tK11VhruYZtS3DiAL6iieTem5AIsB4Fq

Just the same as the free demo, we have provided three kinds of versions of our Workday-Pro-Integrations preparation exam, among which the PDF version is the most popular one. It is understandable that many people give their priority to use paper-based Workday-Pro-Integrations Materials rather than learning on computers, and it is quite clear that the PDF version is convenient for our customers to read and print the contents in our Workday-Pro-Integrations study guide.

Workday Workday-Pro-Integrations Exam Syllabus Topics:

Topic	Details
Topic 1	• 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 2	• 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 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	• 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 5	• 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.

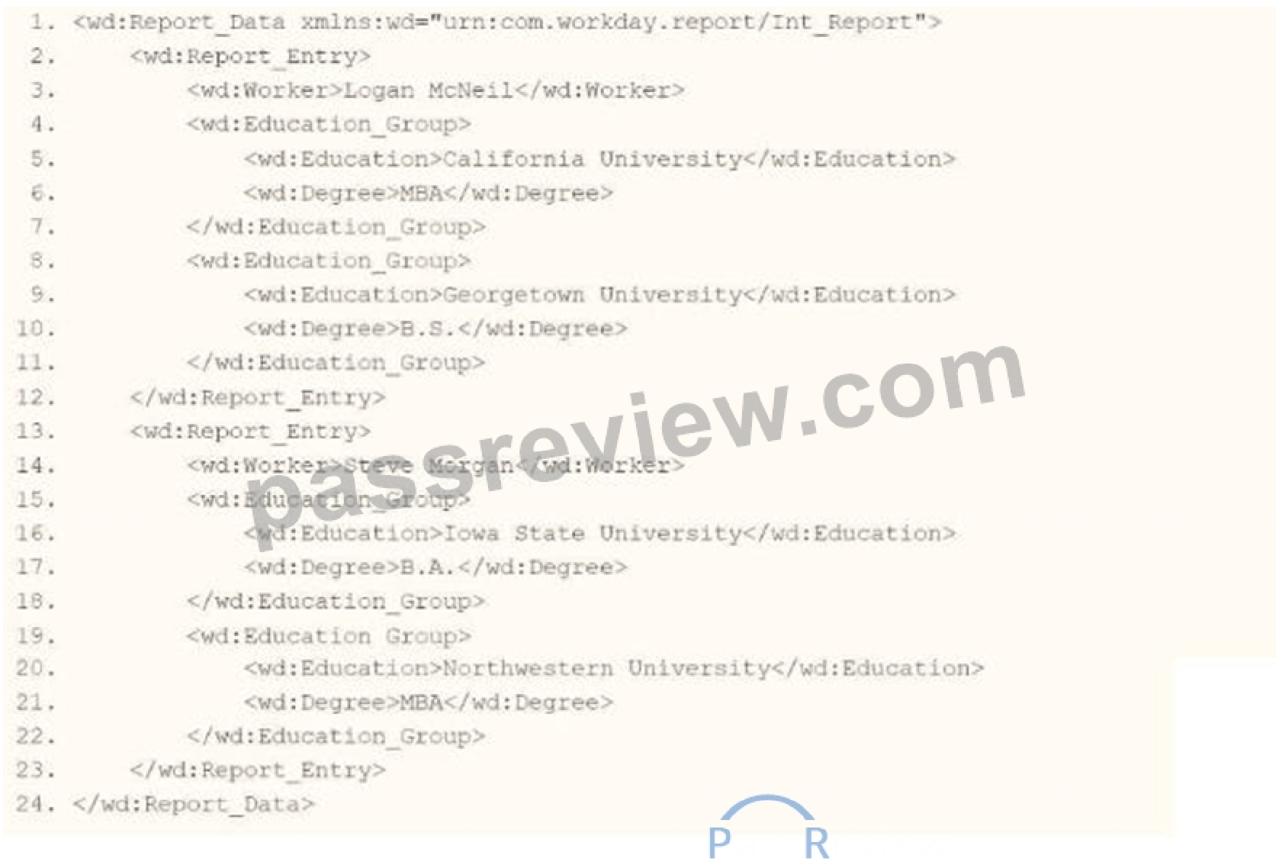
>> Latest Workday-Pro-Integrations Test Testking <<

Reliable Workday Workday-Pro-Integrations Braindumps Sheet | Workday-Pro-Integrations Free Study Material

Our Workday-Pro-Integrations exam training' developers to stand in the perspective of candidate and meet the conditions for each user to tailor their Workday-Pro-Integrations learning materials. What's more, our Workday-Pro-Integrations guide questions are cheap and cheap, and we buy more and deliver more. The more customers we buy, the bigger the discount will be. In order to make the user a better experience to the superiority of our Workday-Pro-Integrations Actual Exam guide, we also provide considerate service, users have any questions related to our Workday-Pro-Integrations study materials, can get the help of our staff in a timely manner.

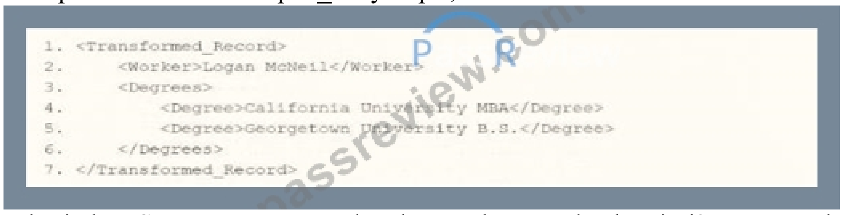
Workday Pro Integrations Certification Exam Sample Questions (Q63-Q68):

NEW QUESTION # 63
Refer to the following XML and example transformed output to answer the question below.



```
<wd:Report_Data xmlns:wd="urn:com.workday.report/Int_Report">
    <wd:Report_Entry>
        <wd:Worker>Logan McNeil</wd:Worker>
        <wd:Education_Group>
            <wd:Education>California University</wd:Education>
            <wd:Degree>MBA</wd:Degree>
        </wd:Education_Group>
        <wd:Education_Group>
            <wd:Education>Georgetown University</wd:Education>
            <wd:Degree>B.S.</wd:Degree>
        </wd:Education_Group>
    </wd:Report_Entry>
    <wd:Report_Entry>
        <wd:Worker>Steve Morgan</wd:Worker>
        <wd:Education_Group>
            <wd:Education>Iowa State University</wd:Education>
            <wd:Degree>B.A.</wd:Degree>
        </wd:Education_Group>
        <wd:Education_Group>
            <wd:Education>Northwestern University</wd:Education>
            <wd:Degree>MBA</wd:Degree>
        </wd:Education_Group>
    </wd:Report_Entry>
</wd:Report_Data>
```

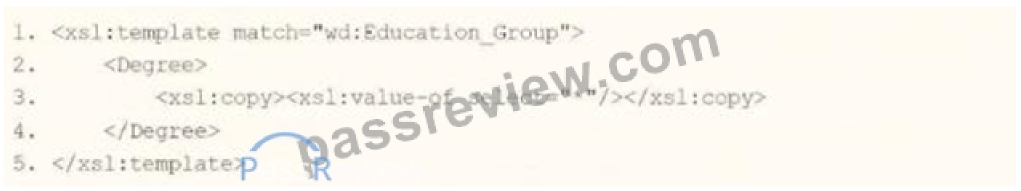
Example transformed wd:Report_Entry output;



```
<Transformed_Record>
    <Worker>Logan McNeil</Worker>
    <Degrees>
        <Degree>California University MBA</Degree>
        <Degree>Georgetown University B.S.</Degree>
    </Degrees>
</Transformed_Record>
```

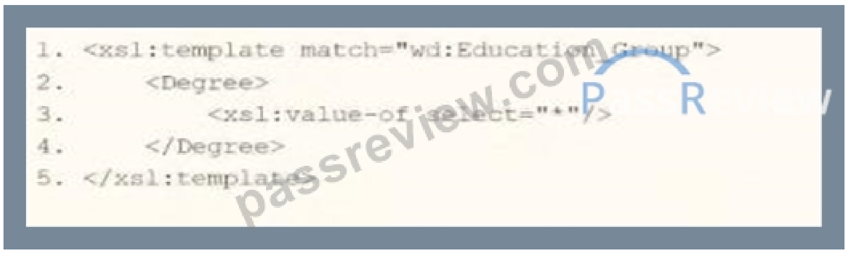
What is the XSLT syntax tor a template that matches on wd: Educationj3roup to produce the degree data in the above Transformed_Record example?

- A.



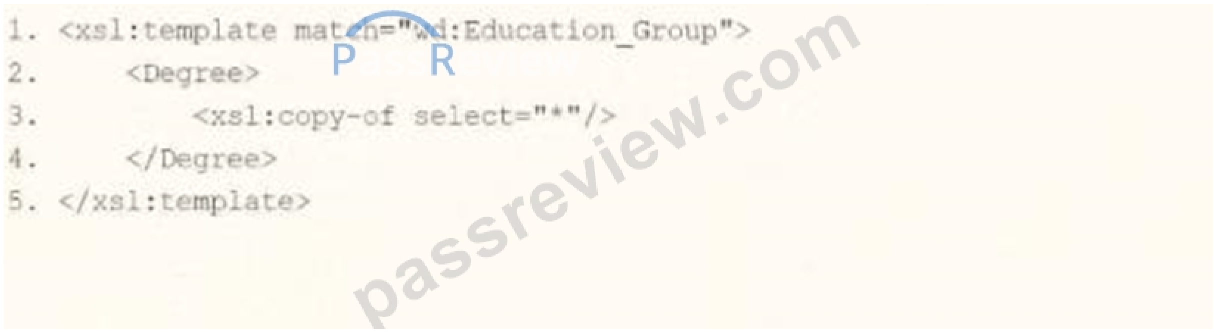
```
<xsl:template match="wd:Education_Group">
    <Degree>
        <xsl:copy><xsl:value-of select="*"/></xsl:copy>
    </Degree>
</xsl:template>
```

- B.

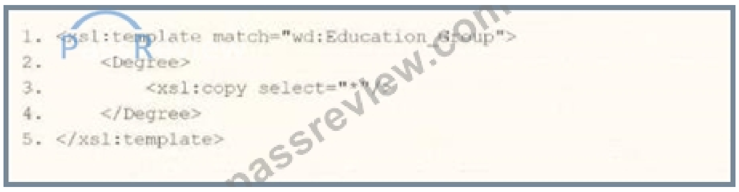


```
<xsl:template match="wd:Education_Group">
    <Degree>
        <xsl:value-of select="*"/>
    </Degree>
</xsl:template>
```

- C.



```
<xsl:template match="wd:Education_Group">
    <Degree>
        <xsl:copy-of select="*"/>
    </Degree>
</xsl:template>
```

- D. 

```
<xsl:template match="wd:Education_Group">
    <Degree>
        <xsl:copy select="*"/>
    </Degree>
</xsl:template>
```

Answer: A

Explanation:
In Workday integrations, XSLT is used to transform XML data, such as the output from a web service-enabled report or EIB, into a desired format for third-party systems. In this scenario, you need to create an XSLT template that matches the wd:Education_Group element in the provided XML and transforms it to produce the degree data in the format shown in the Transformed_Record example. The goal is to output each degree (e.g., "California University MBA" and "Georgetown University B.S.") as a <Degree> element within a <Degrees> parent element.
Here's why option A is correct:
Template Matching: The <xsl:template match="wd:Education_Group"> correctly targets the wd:Education_Group element in the XML, which contains multiple wd:Education elements, each with a wd:Degree child, as shown in the XML snippet (e.g., <wd:Education>California University</wd:Education><wd:Degree>MBA</wd:Degree>).
Transformation Logic:
<Degree> creates the outer <Degree> element for each education group, matching the structure in the Transformed_Record example (e.g., <Degree>California University MBA</Degree>).
<xsl:copy><xsl:value-of select="*"/></xsl:copy> copies the content of the child elements (wd:Education and wd:Degree) and concatenates their values into a single string. The select="*" targets all child elements of wd:Education_Group, and xsl:value-of outputs their text content (e.g., "California University" and "MBA" become "California University MBA").
This approach ensures that each wd:Education_Group is transformed into a single <Degree> element with the combined text of the wd:Education and wd:Degree values, matching the example output.
Context and Output: The template operates on each wd:Education_Group, producing the nested structure shown in the Transformed_Record (e.g., <Degrees><Degree>California University MBA</Degree><Degree>Georgetown University B.S.</Degree></Degrees>), assuming a parent template or additional logic wraps the <Degree> elements in <Degrees>.
Why not the other options?
B .
xml
WrapCopy
<xsl:template match="wd:Education_Group">
<Degree>
<xsl:value-of select="*"/>
</Degree>
</xsl:template>
This uses <xsl:value-of select="*"/> without <xsl:copy>, which outputs the concatenated text of all child elements but does not preserve any XML structure or formatting. It would produce plain text (e.g., "California UniversityMBACalifornia UniversityB.S.") without the proper <Degree> tags, failing to match the structured output in the example.
C .
xml
WrapCopy
<xsl:template match="wd:Education_Group">
<Degree>
<xsl:copy select="*"/>
</Degree>
</xsl:template>
This uses <xsl:copy select="*"/>, but <xsl:copy> does not take a select attribute-it simply copies the current node. This would result

in an invalid XSLT syntax and fail to produce the desired output, making it incorrect.
D .
xml
WrapCopy
<xsl:template match="wd:Education_Group">
<Degree>
<xsl:copy-of select="*"/>
</Degree>
</xsl:template>
This uses <xsl:copy-of select="*"/>, which copies all child nodes (e.g., wd:Education and wd:Degree) as-is, including their element structure, resulting in output like <Degree><wd:Education>California University</wd:Education><wd:Degree>MBA</wd:Degree></Degree>. This does not match the flattened, concatenated text format in the Transformed_Record example (e.g., <Degree>California University MBA</Degree>), making it incorrect.
To implement this in XSLT for a Workday integration:
Use the template from option A to match wd:Education_Group, apply <xsl:copy><xsl:value-of select="*"/></xsl:copy> to concatenate and output the wd:Education and wd:Degree values as a single <Degree> element. This ensures the transformation aligns with the Transformed_Record example, producing the required format for the integration output.
:
Workday Pro Integrations Study Guide: Section on "XSLT Transformations for Workday Integrations" - Details the use of <xsl:template>, <xsl:copy>, and <xsl:value-of> for transforming XML data, including handling grouped elements like wd:Education_Group.
Workday EIB and Web Services Guide: Chapter on "XML and XSLT for Report Data" - Explains the structure of Workday XML (e.g., wd:Education_Group, wd:Education, wd:Degree) and how to use XSLT to transform education data into a flattened format.
Workday Reporting and Analytics Guide: Section on "Web Service-Enabled Reports" - Covers integrating report outputs with XSLT for transformations, including examples of concatenating and restructuring data for third-party systems.

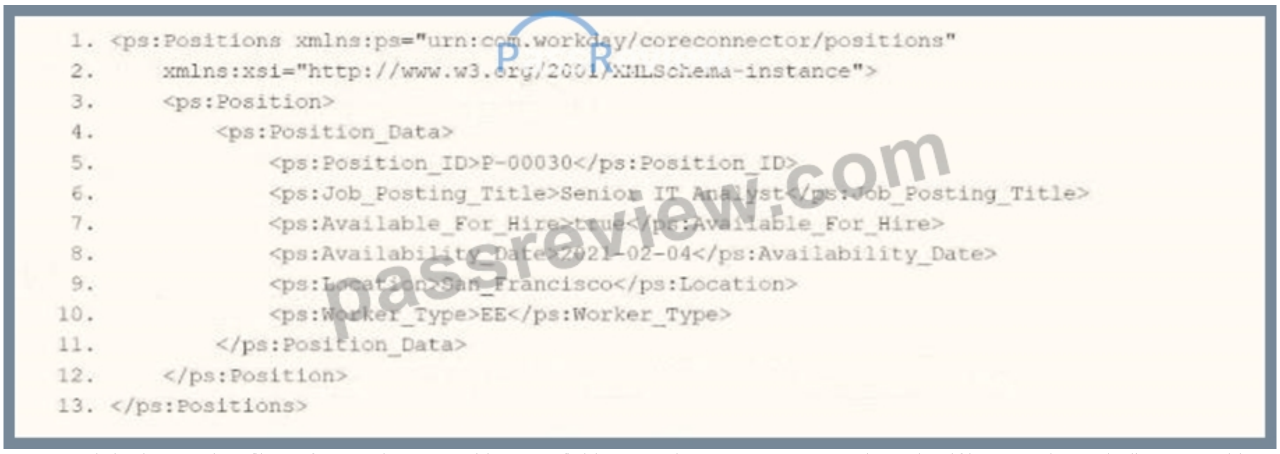
NEW QUESTION # 64
How do you initially upload the XSLT file to a Document Transformation integration system?

- A. In the Global Workday Search bar, run the Edit Integration Attachment Service task.
- B. In the Global Workday Search bar, run the Edit Integration Service Attachment task.
- C. From the Related Action on the Document Transformation, select Configure Integration Attachment Service.
- D. From the Related Action on the Document Transformation, select Configure Integration Attributes.

Answer: C

Explanation:
To upload an XSLT file to a Document Transformation integration system, you use the Configure Integration Attachment Service.
As per Workday documentation:
"The Configure Integration Attachment Service option on the Related Actions menu allows you to attach and manage XSLT files or other transformation documents used in Document Transformation integrations." This is the initial and correct method to upload the XSLT used for transforming incoming or outgoing XML.
Why the others are incorrect:
* B. Configure Integration Attributes configures integration behavior, not attachments.
* C and D reference invalid or misnamed tasks; they are not valid Workday tasks for XSLT upload.
Reference:Workday Pro: Document Transformation Integration Guide - "Uploading and managing XSLT via Configure Integration Attachment Service"

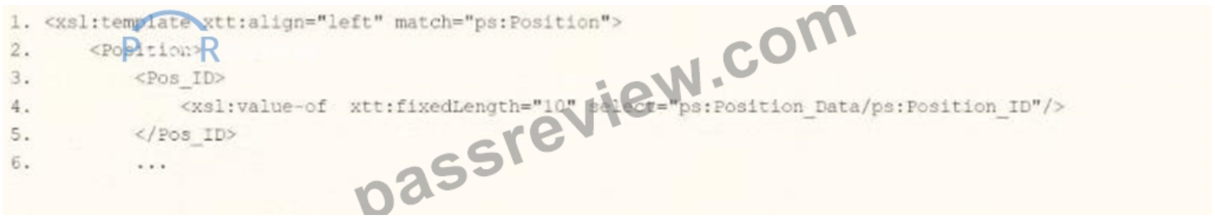
NEW QUESTION # 65
Refer to the following XML data source to answer the question below.



```
<ps:Positions xmlns:ps="urn:com.workday/coreconnector/positions"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ps:Position>
        <ps:Position_Data>
            <ps:Position_ID>P-00030</ps:Position_ID>
            <ps:Job_Posting_Title>Senior IT Analyst</ps:Job_Posting_Title>
            <ps:Available_For_Hire>true</ps:Available_For_Hire>
            <ps:Availability_Date>2021-02-04</ps:Availability_Date>
            <ps:Location>San_Francisco</ps:Location>
            <ps:Worker_Type>EE</ps:Worker_Type>
        </ps:Position_Data>
    </ps:Position>
</ps:Positions>
```

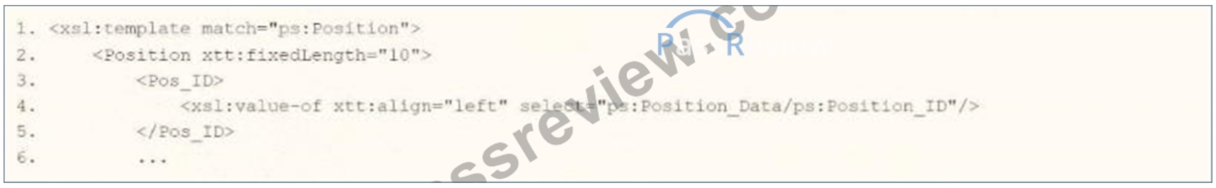
You need the integration file to format the ps:Position_ID field to 10 characters, truncate the value if it exceeds, and align everything to the left.
How will you start your template match on ps:Position to use Document Transformation (DT) to do the transformation using XTT?

- A.



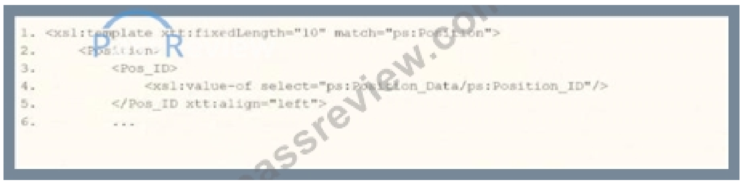
```
<xsl:template xtt:align="left" match="ps:Position">
    <Position>
        <Pos_ID>
            <xsl:value-of  xtt:fixedLength="10" select="ps:Position_Data/ps:Position_ID"/>
        </Pos_ID>
        ...
```

- B.



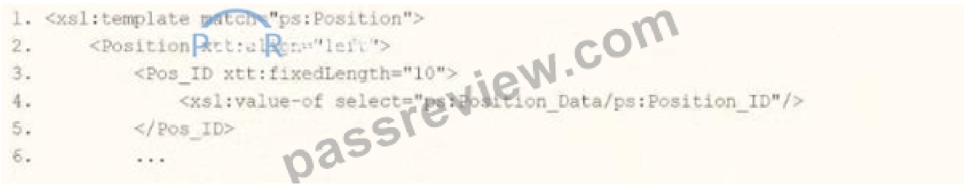
```
<xsl:template match="ps:Position">
    <Position xtt:fixedLength="10">
        <Pos_ID>
            <xsl:value-of xtt:align="left" select="ps:Position_Data/ps:Position_ID"/>
        </Pos_ID>
        ...
```

- C.



```
<xsl:template xtt:fixedLength="10" match="ps:Position">
    <Position>
        <Pos_ID>
            <xsl:value-of select="ps:Position_Data/ps:Position_ID"/>
        </Pos_ID xtt:align="left">
        ...
```

- D.



```
<xsl:template match="ps:Position">
    <Position xtt:align="left">
        <Pos_ID xtt:fixedLength="10">
            <xsl:value-of select="ps:Position_Data/ps:Position_ID"/>
        </Pos_ID>
        ...
```

Answer: D

Explanation:
In Workday integrations, Document Transformation (DT) using XSLT with Workday Transformation Toolkit (XTT) attributes is used to transform XML data, such as the output from a Core Connector or EIB, into a specific format for third-party systems. In this scenario, you need to transform the ps:Position_ID field within the ps:Position element to a fixed length of 10 characters, truncate the value if it exceeds 10 characters, and align the output to the left. The template must match the ps:Position element and apply these formatting rules using XTT attributes.
Here's why option A is correct:
Template Matching: The <xsl:template match="ps:Position"> correctly targets the ps:Position element in the XML, as shown in the provided snippet, ensuring the transformation applies to the appropriate node.
XTT Attributes:
xtt:fixedLength="10" specifies that the Pos_ID field should be formatted to a fixed length of 10 characters. If the ps:Position_ID value exceeds 10 characters, it will be truncated (by default, XTT truncates without raising an error unless explicitly configured otherwise), meeting the requirement to truncate if the value exceeds.
xtt:align="left" ensures that the output is left-aligned within the 10-character field, aligning with the requirement to align everything to

the left.
XPath Selection: The <xsl:value-of select="ps:Position_Data/ps:Position_ID"/> correctly extracts the ps:Position_ID value (e.g., "P-00030") from the ps:Position_Data child element, as shown in the XML structure.
Output Structure: The <Position><Pos_ID>...</Pos_ID></Position> structure ensures the transformed data is wrapped in meaningful tags for the target system, maintaining consistency with Workday integration practices.
Why not the other options?
B .
xml
WrapCopy
```
<xsl:template xtt:align="left" match="ps:Position">
<Position>
<Pos_ID xtt:fixedLength="10">
<xsl:value-of select="ps:Position_Data/ps:Position_ID"/>
</Pos_ID>
</Position>
</xsl:template>
```
This applies xtt:align="left" to the xsl:template element instead of the Pos_ID element. XTT attributes like fixedLength and align must be applied directly to the element being formatted (Pos_ID), not the template itself, making this incorrect.
C .
xml
WrapCopy
```
<xsl:template match="ps:Position">
<Position xtt:fixedLength="10">
<Pos_ID xtt:align="left">
<xsl:value-of select="ps:Position_Data/ps:Position_ID"/>
</Pos_ID>
</Position>
</xsl:template>
```
This applies xtt:fixedLength="10" to the Position element and xtt:align="left" to Pos_ID. However, XTT attributes like fixedLength and align should be applied to the specific field being formatted (Pos_ID), not the parent element (Position). This misplacement makes it incorrect.
D .
xml
WrapCopy
```
<xsl:template xtt:fixedLength="10" match="ps:Position">
<Position>
<Pos_ID xtt:align="left">
<xsl:value-of select="ps:Position_Data/ps:Position_ID"/>
</Pos_ID>
</Position>
</xsl:template>
```
This applies xtt:fixedLength="10" to the xsl:template element and xtt:align="left" to Pos_ID. Similar to option B, XTT attributes must be applied to the specific element (Pos_ID) being formatted, not the template itself, making this incorrect.
To implement this in XSLT for a Workday integration:
Use the template from option A to match ps:Position, apply xtt:fixedLength="10" and xtt:align="left" to the Pos_ID element, and extract the ps:Position_ID value using the correct XPath. This ensures the ps:Position_ID (e.g., "P-00030") is formatted to 10 characters, truncated if necessary, and left-aligned, meeting the integration file requirements.
:
Workday Pro Integrations Study Guide: Section on "Document Transformation (DT) and XTT" - Details the use of XTT attributes like fixedLength and align for formatting data in XSLT transformations, including truncation behavior.
Workday Core Connector and EIB Guide: Chapter on "XML Transformations" - Explains how to use XSLT templates with XTT attributes to transform position data, including fixed-length formatting and alignment.
Workday Integration System Fundamentals: Section on "XTT in Integrations" - Covers the application of XTT attributes to specific fields in XML for integration outputs, ensuring compliance with formatting requirements like length and alignment.

NEW QUESTION # 66
Refer to the following scenario to answer the question below.
You have been asked to build an integration using the Core Connector: Worker template and should leverage the Data Initialization Service (DIS). The integration will be used to export a full file (no change detection) for employees only and will include personal

data.
What configuration is required to output the value of a calculated field which you created for inclusion in this integration?

- A. Configure Integration Field Attributes.
- B. Configure Integration Maps.
- C. Configure Integration Field Overrides.
- D. Configure Integration Attributes.

Answer: C

Explanation:
The scenario involves a Core Connector: Worker integration using the Data Initialization Service (DIS) to export a full file of employee personal data, with a requirement to include a calculated field in the output.
Core Connectors rely on predefined field mappings, but custom calculated fields need specific configuration to be included. Let's analyze the solution:
* Requirement:Output the value of a calculated field created for this integration. In Workday, calculated fields are custom-built (e.g., using Report Writer or Calculated Fields) and not part of the standard Core Connector template, so they must be explicitly added to the output.
* Integration Field Overrides:In Core Connectors, Integration Field Overrides allow you to replace a delivered field's value or add a new field to the output by mapping it to a calculated field. This is the standard method to include custom calculated fields in the integration file. You create the calculated field separately, then use overrides to specify where its value appears in the output structure (e.g., as a new column or replacing an existing field).
* Option Analysis:
* A. Configure Integration Field Attributes: Incorrect. Integration Field Attributes refine how delivered fields are output (e.g., filtering multi-instance data like phone type), but they don't support adding or mapping calculated fields.
* B. Configure Integration Field Overrides: Correct. This configuration maps the calculated field to the output, ensuring its value is included in the exported file.
* C. Configure Integration Attributes: Incorrect. Integration Attributes define integration-level settings (e.g., file name, delivery protocol), not field-specific outputs like calculated fields.
* D. Configure Integration Maps: Incorrect. Integration Maps transform existing field values (e.g.,
"Married" to "M"), but they don't add new fields or directly output calculated fields.
* Implementation:
* Create the calculated field in Workday (e.g., via Create Calculated Field task).
* Edit the Core Connector: Worker integration.
* Navigate to the Integration Field Overrides section.
* Add a new override, selecting the calculated field and specifying its output position (e.g., a new field ID or overriding an existing one).
* Test the integration to confirm the calculated field value appears in the output file.
References from Workday Pro Integrations Study Guide:
* Core Connectors & Document Transformation: Section on "Configuring Integration Field Overrides" explains how to include calculated fields in Core Connector outputs.
* Integration System Fundamentals: Notes the use of overrides for custom data in predefined integration templates.

NEW QUESTION # 67
What XSL component is required to execute valid transformation instructions in the XSLT code?

- A. xsl:output
- B. xsl:apply-template
- C. xsl:template
- D. xsl:call-template

Answer: C

Explanation:
The <xsl:template> is the core component in XSLT. It defines the transformation rules that will be applied to nodes in the XML document.
"Without at least one <xsl:template> element, an XSLT file cannot perform any transformation. This is the execution block where processing logic begins." Why the others are incorrect:
B . <xsl:apply-templates> applies templates but is not valid without the actual template definitions.
C . <xsl:call-template> calls named templates - which must first exist.

D . <xsl:output> defines format but does not perform transformation logic.

NEW QUESTION # 68

......

For candidates who want to evaluate and enhance their Workday Workday-Pro-Integrations Test Preparation online, the web-based practice test is a perfect choice. You can attempt our 60 Workday web-based practice exam whenever it suits you because it is accessible from any location with an internet connection. This Workday Pro Integrations Certification Exam browser-based practice exam helps you overcome exam fear as it simulates the environment of the real test.

Reliable Workday-Pro-Integrations Braindumps Sheet: https://www.passreview.com/Workday-Pro-Integrations_exam-braindumps.html