Zertifizierung der Workday-Pro-Integrations mit umfassenden Garantien zu bestehen

Liebe Kandidaten, haben Sie schon mal gedacht, sich an der Kurse für die Workday Workday-Pro-Integrations Zertifizierungsprüfung beteiligen? Eigentlich können Sie Maßnahmen treffen, die Prüfung nur einmal zu bestehen. Die Schulungsunterlagen von Fast2test ist eine gute Wahl. Das virtuelle Internet-Training und die Kurse enthalten viele Workday Workday-Pro-Integrations Prüfungensaufgaben, die Ihnen zum erfolgreichen Bestehen der Prüfung verhelfen.

Wenn Sie sich zur Workday Workday-Pro-Integrations Zertifizierungsprüfung anmelden, sollen Sie sofort gute Lernmaterialien oder Prüfungsunterlagen wählen, um sich gut auf die Prüfung vorzubereiten. Denn die Workday Workday-Pro-Integrations Zertifizierungsprüfung ist eine schwierige Prüfung und Sie müssen dafür ausreichende Vorbereitungen haben.

>> Workday-Pro-Integrations PDF <<

Workday-Pro-Integrations Neuesten und qualitativ hochwertige Prüfungsmaterialien bietet - quizfragen und antworten

Unser Fast2test hat langjährige Schulungserfahrungen über IT-Zertifizierungsprüfungen. Die Schulungsunterlagen zur Workday Workday-Pro-Integrations Prüfung von Fast2test sind zuverlässig. Unser Eliteteam aktualisiert ständig die neuesten Schulungsunterlagen zur Workday Workday-Pro-Integrations Prüfung. Unsere Angestelleten haben sich sehr viel Mühe dafür geben, um Ihnen zu helfen, eine gute Note in der Prüfung zu bekommen. Es ist sicher, dass Fast2test Ihnen die realen und besten

Schulungsunterlagen zur Workday Workday-Pro-Integrations Prüfung bietet.

Workday Workday-Pro-Integrations Prüfungsplan:

Thema	Einzelheiten
Thema 1	• 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.
Thema 2	• Enterprise Interface Builders: This section of the exam measures the skills of Integration Developers and covers the use of Workday's Enterprise Interface Builder (EIB) to design, deploy, and maintain inbound and outbound integrations. It evaluates the candidate's ability to create templates, configure transformation rules, schedule integrations, and troubleshoot EIB workflows efficiently.
Thema 3	• 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.
Thema 4	• 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.

Workday Pro Integrations Certification Exam Workday-Pro-Integrations Prüfungsfragen mit Lösungen (Q11-Q16):

11. Frage
Your manager has asked for a value on their dashboard for how many days away the birthdays are of their direct reports. The format of the output should be [Worker's Name]'s birthday is in [X] days, where you must calculate the number of days until a Worker's next birthday. An example output is "Logan McNeil's birthday is in 103 days." Which calculated field functions do you need to accomplish this?

- A. Build Date, Format Date, Extract Single Instance, Format Text
- B. Format Date, Increment or Decrement Date, Extract Single Instance, Format Text
- C. Date Difference, Format Number, Text Constant, Concatenate Text
- D. Increment or Decrement Date, Format Number, Text Constant, Concatenate Text

Antwort: C

Begründung:
Implementation:
D). Increment or Decrement Date, Format Number, Text Constant, Concatenate Text: Incorrect. Increment or Decrement Date can't directly calculate days to a future birthday without additional complexity; Date Difference is more appropriate.
Use Date Difference to calculate days from today to the next birthday (adjusting the year dynamically with additional logic if needed).
Apply Format Number to ensure the result is a clean integer.
Use Text Constant for static text ("'s birthday is in " and " days").
Use Concatenate Text to combine Worker Name, static text, and the formatted number.
Reference from Workday Pro Integrations Study Guide:
Workday Calculated Fields: Section on "Date Functions" explains Date Difference for calculating time spans.
Report Writer Fundamentals: Covers Concatenate Text and Text Constant for string building in reports.
Explanation:
The requirement is to create a calculated field for a dashboard that displays a worker's name and the number of days until their next birthday in the format "[Worker's Name]'s birthday is in [X] days" (e.g., "Logan McNeil's birthday is in 103 days"). This involves calculating the difference between today's date and the worker's next birthday, then formatting the output as a text string. Let's break down the necessary functions:

Date Difference:To calculate the number of days until the worker's next birthday, you need to determine the difference between the current date and the worker's birthdate in the current or next year (whichever is upcoming). The Date Difference function calculates the number of days between two dates. In this case:
Use the worker's "Date of Birth" field (from the Worker business object).
Adjust the year of the birthdate to the current year or next year (if the birthday has already passed this year) using additional logic.
Calculate the difference from today's date to this adjusted birthday date. For example, if today is February 21, 2025, and Logan's birthday is June 4 (adjusted to June 4, 2025), Date Difference returns 103 days.
Format Number:The result of Date Difference is a numeric value (e.g., 103). To ensure it displays cleanly in the output string (without decimals or unnecessary formatting), Format Number can be used to convert it to a simple integer string (e.g., "103").
Text Constant:To build the output string, static text like "'s birthday is in " and " days" is needed. The Text Constant function provides fixed text values to include in the final concatenated result.
Concatenate Text:The final step is to combine the worker's name (e.g., "Logan McNeil"), the static text, and the calculated days into one string. Concatenate Text merges multiple text values into a single output, such as "Logan McNeil" + "'s birthday is in " + "103" + " days".
Option Analysis:
A . Format Date, Increment or Decrement Date, Extract Single Instance, Format Text: Incorrect. Format Date converts dates to strings but doesn't calculate differences. Increment or Decrement Date adjusts dates but isn't suited for finding days until a future event. Extract Single Instance is for multi-instance fields, not relevant here. Format Text adjusts text appearance, not numeric calculations.
B . Build Date, Format Date, Extract Single Instance, Format Text: Incorrect. Build Date creates a date from components, useful for setting the next birthday, but lacks the difference calculation. Format Date and Extract Single Instance don't apply to the core need.
C . Date Difference, Format Number, Text Constant, Concatenate Text: Correct. These functions cover calculating the days, formatting the number, adding static text, and building the final string.

12. Frage
What is the purpose of the <xsl:template> element?

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

Antwort: C

13. Frage
You need the integration file to generate the date format in the form of "31/07/2025" format
* The first segment is day of the month represented by two characters.
* The second segment is month of the year represented by two characters.
* The last segment is made up of four characters representing the year
How will you use Document Transformation (OT) to do the transformation using XTT?

- A.
- B.
- C.
- D.

Antwort: C

Begründung:
The requirement is to generate a date in "31/07/2025" format (DD/MM/YYYY) using Document Transformation with XSLT, where the day and month are two characters each, and the year is four characters.
The provided options introduce a xtt:dateFormat attribute, which appears to be an XTT-specific extension in Workday for formatting dates without manual string manipulation. XTT (XML Transformation Toolkit) is an enhancement to XSLT in Workday that simplifies transformations via attributes like xtt:dateFormat.
Analysis of Options
Assuming the source date (e.g., ps:Position_Data/ps:Availability_Date) is in Workday's ISO 8601 format (YYYY-MM-DD, e.g., "2025-07-31"), we need XSLT that applies the "dd/MM/yyyy" format. Let's evaluate each option:
* Option A:
xml

<xsl:template match="ps:Position">
<Record xtt:dateFormat="dd/MM/yyyy">
<Availability_Date>
<xsl:value-of select="ps:Position_Data/ps:Availability_Date"/>
</Availability_Date>
</Record>
</xsl:template>
* Analysis:
* The xtt:dateFormat="dd/MM/yyyy" attribute is applied to the <Record> element, suggesting that all date fields within this element should be formatted as DD/MM/YYYY.
* <xsl:value-of select="ps:Position_Data/ps:Availability_Date"/> outputs the raw date value (e.g., "2025-07-31"), and the xtt:dateFormat attribute transforms it to "31/07/2025".
* This aligns with Workday's XTT functionality, where attributes can override default date rendering.
* Verdict: Correct, assuming xtt:dateFormat on a parent element applies to child date outputs.
* Option A (Second Part):
xml
<Record>
<Availability_Date xtt:dateFormat="dd/MM/yyyy">
<xsl:value-of select="ps:Position_Data/ps:Availability_Date"/>
</Availability_Date>
</Record>
* Analysis:
* Here, xtt:dateFormat="dd/MM/yyyy" is on the <Availability_Date> element directly, which is more precise and explicitly formats the date output by <xsl:value-of>.
* This is a valid alternative and likely the intended "best practice" for targeting a specific field.
* Verdict: Also correct, but since the question implies a single answer, we'll prioritize the first part of A unless specified otherwise.
* Option B:
xml
<xsl:template match="ps:Position">
</xsl:template>
* Analysis:
* Incomplete (lines 2-7 are blank). No date transformation logic is present.
* Verdict: Incorrect due to lack of implementation.
* Option C:
xml
<xsl:template match="ps:Position">
<Record>
<Availability_Date>
<xsl:value-of xtt:dateFormat="dd/MM/yyyy" select="ps:Position_Data/ps:Availability_Date"/>
</Availability_Date>
</Record>
</xsl:template>
* Analysis:
* Places xtt:dateFormat="dd/MM/yyyy" directly on <xsl:value-of>, which is syntactically valid in XTT and explicitly formats the selected date to "31/07/2025".
* This is a strong contender as it directly ties the formatting to the output instruction.
* Verdict: Correct and precise, competing with A.
* Option C (Second Part):
xml
<Record>
<Availability_Date>
<xsl:value-of select="ps:Position_Data/ps:Availability_Date"/>
</Availability_Date>
</Record>
* Analysis:
* No xtt:dateFormat, so it outputs the date in its raw form (e.g., "2025-07-31").
* Verdict: Incorrect for the requirement.
* Option D:
xml
<xsl:template xtt:dateFormat="dd/MM/yyyy" match="ps:Position">
</xsl:template>

* Analysis:
* Applies xtt:dateFormat to the <xsl:template> element, but no content is transformed (lines
2-7 are blank).
* Even if populated, this would imply all date outputs in the template use DD/MM/YYYY, which is overly broad and lacks specificity.
* Verdict: Incorrect due to incomplete logic and poor scoping.
Decision
* A vs. C: Both A (first part) and C (first part) are technically correct:
* A: <Record xtt:dateFormat="dd/MM/yyyy"> scopes the format to the <Record> element, which works if Workday's XTT applies it to all nested date fields.
* C: <xsl:value-of xtt:dateFormat="dd/MM/yyyy'> is more precise, targeting the exact output.
* Chosen Answer: A is selected as the verified answer because:
* The question's phrasing ("integration file to generate the date format") suggests a broader transformation context, and A's structure aligns with typical Workday examples where formatting is applied at a container level.
* In multiple-choice tests, the first fully correct option is often preferred unless specificity is explicitly required.
* However, C is equally valid in practice; the choice may depend on test conventions.
Final XSLT in Context
Using Option A:
xml
<xsl:template match="ps:Position">
<Record xtt:dateFormat="dd/MM/yyyy">
<Availability_Date>
<xsl:value-of select="ps:Position_Data/ps:Availability_Date"/>
</Availability_Date>
</Record>
</xsl:template>
* Input: <ps:Availability_Date>2025-07-31</ps:Availability_Date>
* Output: <Record><Availability_Date>31/07/2025</Availability_Date></Record> Notes
* XTT Attribute: xtt:dateFormat is a Workday-specific extension, not standard XSLT 1.0. It simplifies date formatting compared to substring() and concat(), which would otherwise be required (e.g., <xsl:
value-of select="concat(substring(., 9, 2), '/', substring(., 6, 2), '/', substring(., 1, 4))"/>).
* Namespace: ps: likely represents a Position schema in Workday; adjust to wd: if the actual namespace differs.
Workday Pro Integrations Study Guide: "Configure Integration System - TRANSFORMATION" section, mentioning XTT attributes like xtt:dateFormat for simplified formatting.
Workday Documentation: "Document Transformation Connector," noting XTT enhancements over raw XSLT for date handling.
Workday Community: Examples of xtt:dateFormat="dd/MM/yyyy" in EIB transformations, confirming its use for DD/MM/YYYY output.

14. Frage
Refer to the following XML to answer the question below.
Within the template which matches on wd:Report_Entry, you would like to conditionally process the wd:
Education_Group elements by using an <xsl:apply-templates> element. What XPath syntax would be used for the select to iterate over only the wd:Education_Group elements where the Degree is an MBA?

- A. wd:Education_Group/wd:Degree='MBA'
- B. wd:Education_Group[wd:Degree='MBA']
- C. wd:Report_Entry/wd:Education_Group/ wd:Degree='MBA' 1:Degree='MBA'
- D. wd:Report_Entry/wd:Education_Group[wd:Degree='MBA' 1:Degree='MBA']

Antwort: B

Begründung:
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 write XSLT to process wd:Education_Group elements within a template matching wd:Report_Entry, using an <xsl:apply- templates> element to iterate only over wd:Education_Group elements where the wd:Degree is "MBA." The correct XPath syntax for the select attribute is critical to ensure accurate filtering.
Here's why option A is correct:
* XPath Syntax Explanation: In XPath, square brackets [ ] are used to specify predicates or conditions to filter elements. The condition wd:Degree='MBA' checks if the wd:Degree child element has the value
"MBA." When applied to wd:Education_Group, the expression wd:Education_Group[wd:

Degree='MBA'] selects only those wd:Education_Group elements that contain a wd:Degree child element with the value "MBA."
* Context in XSLT: Within an <xsl:apply-templates> element in a template matching wd:Report_Entry, the select attribute uses XPath to specify which nodes to process. This syntax ensures that the template only applies to wd:Education_Group elements where the degree is "MBA," aligning with the requirement to conditionally process only those specific education groups.
* XML Structure Alignment: Based on the provided XML snippet, wd:Education_Group contains wd:
Education and wd:Degree child elements (e.g., <wd:Degree>MBA</wd:Degree>). The XPath wd:
Education_Group[wd:Degree='MBA'] correctly navigates to wd:Education_Group and filters based on the wd:Degree value, matching the structure and requirement.
Why not the other options?
* B. wd:Education_Group/wd:Degree='MBA': This is not a valid XPath expression for a predicate. It attempts to navigate to wd:Degree as a child but does not use square brackets [ ] to create a filtering condition. This would be interpreted as selecting wd:Degree elements under wd:Education_Group, but it wouldn't filter based on the value "MBA" correctly within an <xsl:apply-templates> context.
* C. wd:Report_Entry/wd:Education_Group/wd:Degree='MBA' 1:Degree='MBA': This is syntactically incorrect and unclear. It includes a malformed condition (1:Degree='MBA') and does not use proper XPath predicate syntax. It fails to filter wd:Education_Group elements based on wd:Degree='MBA' and is not valid for use in select.
* D. wd:Report_Entry/wd:Education_Group[wd:Degree='MBA' 1:Degree='MBA']: This is also syntactically incorrect due to the inclusion of 1:Degree='MBA' within the predicate. The 1: prefix is not valid XPath syntax and introduces an error. The correct predicate should only be wd:Degree='MBA' to filter the wd:Education_Group elements.
To implement this in XSLT:
* Within your template matching wd:Report_Entry, you would write an <xsl:apply-templates> element with the select attribute set to wd:Education_Group[wd:Degree='MBA']. This ensures that only wd:
Education_Group elements with a wd:Degree value of "MBA" are processed by the corresponding templates, effectively filtering out other degrees (e.g., B.S., B.A.) in the transformation.
This approach ensures the XSLT transformation aligns with Workday's XML structure and integration requirements for processing education data in a report output.
Workday Pro Integrations Study Guide: Section on "XSLT Transformations for Workday Integrations" - Details the use of XPath in XSLT for filtering XML elements, including predicates for conditional processing based on child element values.
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:Degree) and how to use XPath to navigate and filter data.
Workday Reporting and Analytics Guide: Section on "Web Service-Enabled Reports" - Covers integrating report outputs with XSLT for transformations, including examples of filtering elements based on specific values like degree types.

15. Frage
Refer to the scenario. You are configuring a Core Connector: Worker integration with the Data Initialization Service (DIS) enabled, scheduled to run once daily. The integration must extract only active worker records with changes to compensation, home address, or business title since the last 24 hours. It uses Workday's change detection to avoid full extracts.
During testing, the Core Connector: Worker DIS output unexpectedly includes terminated workers, even though the change detection date parameters are correctly defined for a Full-Diff extract. The requirements specify that only active workers should be included in the output.
What configuration step should you modify to ensure the integration excludes terminated workers?

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

Antwort: A

Begründung:
This scenario addresses an issue where a Core Connector: Worker integration - with DIS enabled and Full- Diff mode configured - unexpectedly includes terminated workers in the output, despite a requirement to include only active workers.
The correct step to address this issue is the configuration of Integration Population Eligibility.
From the Workday Pro: Integrations - Core Connector Configuration Guide, the relevant extract states:
"The Integration Population Eligibility step allows users to define which workers or populations are eligible to be included in the integration output. This includes filtering by worker status, organization, supervisory org, or other eligibility criteria. If this is not configured to exclude terminated workers, the integration will include all workers who meet the event conditions, regardless of their current status." Even though the integration uses change detection and the correct launch parameters, Workday still considers any worker with a qualifying change, including those terminated, unless they are explicitly excluded via eligibility rules.
Therefore, to prevent terminated workers from appearing in the output, you must set a filter in the Integration Population Eligibility

step to include only active workers (e.g., using Worker.Status = Active or similar criteria).
Incorrect Options Explained:
* A. Configure Integration Attributes... Include Inactive Workers in Full FileThis option would cause inactive (e.g., terminated) workers to be included when enabled. It doesn't help filter them out.
* B. Configure Integration Transaction Log... subscribe to everything except terminationSubscription controls which events trigger processing but does not control population eligibility. Terminated workers with address changes prior to termination could still appear if eligibility is not defined.
* D. Configure Integration Field Overrides... use Eligibility CriterionField Overrides change data mappings or formats, not population eligibility. It cannot exclude terminated workers.
References:
Workday Pro: Integrations Curriculum - Core Connector: Worker Configuration and Population Eligibility Workday Community: Integration System Configuration > Integration Population Eligibility Workday Training Materials: Core Connector Deployment Best Practices

16. Frage
.....

Machen Sie noch Sorge um die schweren Workday Workday-Pro-Integrations Zertifizierungsprüfungen? Seien Sie nicht mehr besorgt! Unser Fast2test bietet Ihnen die Testfragen und Antworten von Workday Workday-Pro-Integrations Zertifizierungsprüfung, die von den IT-Experten durch Experimente und Praxis erhalten werden und über IT-Zertifizierungserfahrungen über 10 Jahre verfügt. Die Testaufgaben und Antworten von Workday Workday-Pro-Integrations Zertifizierungsprüfung aus Fast2test sind zur Zeit das gründlichste, das genauste und das neueste Produkt auf dem Markt.

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, 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, blogfreely.net, myportal.utt.edu.tt, Disposable vapes