

Workday-Pro-Integrationsテスト模擬問題集 & Workday-Pro-Integrations試験復習



さらに、Topexam Workday-Pro-Integrationsダンプの一部が現在無料で提供されています：<https://drive.google.com/open?id=1qNfZvxjTlazNpvlWzm0R4Q8Aqa5ekmFo>

弊社のWorkday-Pro-Integrations問題集はIT業界で有名で、ブランドになっています。Workday-Pro-Integrations問題集はすごく人気がある商品で、どこでも広告を掲載する必要がないです。従って、多くの受験者は弊社のWorkday-Pro-Integrations問題集を選びました。なぜ彼らがWorkday-Pro-Integrations問題集を選ぶかというと、弊社のWorkday-Pro-Integrations問題集は高品質で、便利で、勉強しやすいからです。弊社のWorkday-Pro-Integrations問題集を買う人は全部Workday-Pro-Integrations試験にいい成績で合格しました。

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

トピック	出題範囲
トピック 1	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.
トピック 2	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.
トピック 3	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.

Workday-Pro-Integrations試験復習 & Workday-Pro-Integrations無料ダウンロード

一方で、Workday-Pro-Integrationsテストトレントは、シラバスの変更および理論と実践の最新の進展に応じて改訂および更新されます。一方、Workday-Pro-Integrationsテスト回答のシンプルで理解しやすい言語は、学習者を学習の困難から解放します-あなたが学生であろうとスタッフであろうと。Workday-Pro-Integrationsガイドトレントの支払いが成功すると、5~10分以内にシステムからメールが届きます。リンクをクリックしてログインすると、すぐにWorkday-Pro-Integrationsガイド急流で学習できます。

Workday Pro Integrations Certification Exam 認定 Workday-Pro-Integrations 試験問題 (Q51-Q56):

質問 # 51

You are configuring integration security for a Core Connector integration system. How do you find the web service operation used by the connector template?

- A. Run the Integration Template Catalog report in the tenant
- B. View the SOAP API Reference on Workday Community
- C. It is displayed when selecting a Core Connector Template to build an integration system
- D. Run the integration system and view the web service request in the messages audit

正解: A

解説:

When setting up security for a Core Connector integration system in Workday, you need to know which web service operation the connector template uses. The best way is to run the "Integration Template Catalog report" within your Workday tenant. This report lists all integration templates and should include details about the web service operations they use, making it easy to configure security.

Why This Matters

This method is efficient because it lets you find the information before running the system, which is crucial for setting up permissions correctly. It's surprising that such a specific report exists, as it simplifies a task that could otherwise involve running the system or guessing from API references.

How It Works

- * Select the report in your Workday tenant to see a list of all Core Connector templates.
- * Look for the template you're using and find the associated web service operation listed in the report.
- * Use this information to set up the right security permissions for your integration.

For more details, check out resources like [Workday Core Connectors](#) or [Workday Integrations](#).

質問 # 52

Refer to the following XML to answer the question below.

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 a template against <wd:Job_Profile>.

What XPath syntax would be used to select the value of the wd:Job_Code element when the <xsl:value-of> element is placed within

the template which matches on <wd:Job_Profile>?

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

正解: C

解説:

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:

Job_Code> element. The root template of your XSLT matches on <wd:Get_Job_Profiles_Response> and applies a template to <wd:Job_Profile>. Within this template, you use the <xsl:value-of> element to extract the <wd:Job_Code> 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 are:
 - * <wd:Job_Profile_Reference>, which contains <wd:ID> elements (e.g., a Job_Profile_ID).
 - * <wd:Job_Profile_Data>, which contains <wd:Job_Code> with the value

Senior_Benefits_Analyst.

The task is to select the value of <wd:Job_Code> (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 <wd:Job_Code>.

Analysis of Options

Let's evaluate each option based on the XML structure and XPath syntax rules:

- * Option A: wd:Job_Profile/wd:Job_Profile_Data/wd:Job_Code
- * This XPath starts from wd:Job_Profile and navigates to wd:Job_Profile_Data/wd:Job_Code.

However, in the XML, <wd:Job_Profile> is the parent element, and <wd:Job_Profile_Data> is a direct child containing <wd:Job_Code>. The path wd:Job_Profile/wd:Job_Profile_Data/wd:

Job_Code is technically correct in terms of structure, as it follows the hierarchy:

- * <wd:Job_Profile> # <wd:Job_Profile_Data> # <wd:Job_Code>.
- * However, since the template matches <wd:Job_Profile>, the context node is already <wd:

Job_Profile>. You don't need to include wd:Job_Profile/ at the beginning of the XPath unless navigating from a higher level. Starting directly with wd:Job_Profile_Data/wd:Job_Code (Option C) is more concise and appropriate for the context. This option is technically valid but redundant and less efficient, making it less preferred compared to Option C.

- * Option B: wd:Job_Profile_Data[@wd:Job_Code]

* This XPath uses an attribute selector ([@wd:Job_Code]) to filter <wd:Job_Profile_Data> based on an attribute named wd:Job_Code. However, examining the XML, <wd:Job_Profile_Data> does not have a wd:Job_Code attribute-it has a child element <wd:Job_Code> with the value

"Senior_Benefits_Analyst." The [@attribute] syntax is used for attributes, not child elements, so this XPath is incorrect. It would not select the <wd:Job_Code> value and would likely return no results or an error. This option is invalid.

- * Option C: wd:Job_Profile_Data/wd:Job_Code

* This XPath starts from wd:Job_Profile_Data (a direct child of <wd:Job_Profile>) and navigates to wd:Job_Code. Since the template matches <wd:Job_Profile>, the contextnode is <wd:

Job_Profile>, and wd:Job_Profile_Data/wd:Job_Code correctly points to the <wd:Job_Code> element within <wd:Job_Profile_Data>. This path is:

- * Concise and appropriate for the context.
- * Directly selects the value "Senior_Benefits_Analyst" when used with <xsl:value-of>.
- * Matches the XML structure, as <wd:Job_Profile_Data> contains <wd:Job_Code> as a child.
- * This is the most straightforward and correct option for selecting the <wd:Job_Code> value within the <wd:Job_Profile> template.
- * Option D: wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID']
- * This XPath navigates to <wd:Job_Profile_Reference> (a child of <wd:Job_Profile>) and then to <wd:ID> with an attribute wd:type="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 XPath wd:Job_Profile_Reference/wd:ID[@wd:type='Job_Profile_ID'] selects the <wd:ID> element with wd:type="Job_Profile_ID", which has the value "Senior_Benefits_Analyst." However, this is not the <wd:Job_Code> value-the <wd:Job_Code> is a separate element under

<wd:Job_Profile_Data>, not <wd:Job_Profile_Reference>. The question specifically asks for the <wd:Job_Code> value, so this option is incorrect, as it selects a different piece of data (the job profile ID, not the job code).

Why Option C is Correct

Option C, wd:Job_Profile_Data/wd:Job_Code, 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_Data/wd:Job_Code>, which directly selects the <wd:Job_Code> element's value ("Senior_Benefits_Analyst").
- * It is concise and aligns with standard XPath navigation in XSLT, avoiding unnecessary redundancy (unlike Option A) or incorrect attribute selectors (unlike Option B).
- * It matches the XML structure, where <wd:Job_Profile_Data> is a child of <wd:Job_Profile> and contains <wd:Job_Code> as a child.

* When used with `<xsl:value-of select='wd:Job_Profile_Data/wd:Job_Code'/>` in the template, it outputs the job code value, fulfilling the requirement.

Practical Example in XSLT

Here's how this might look in your XSLT:

xml

WrapCopy

```
<xsl:template match='wd:Job_Profile'>
```

```
<xsl:value-of select='wd:Job_Profile_Data/wd:Job_Code'/>
```

```
</xsl:template>
```

This would output "Senior_Benefits_Analyst" for the `<wd:Job_Code>` element 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_Data>` as the container for job profile details, including `<wd:`

`Job_Code>`. The guide emphasizes using relative XPath paths within templates to navigate from the matched element (e.g.,

`<wd:Job_Profile>`) to child elements like `<wd:Job_Profile_Data/wd:Job_Code>`.

Workday Pro Integrations Study Guide References

* Section: XSLT Transformations in EIBs- Describes using XSLT to transform web service responses, including selecting elements with XPath.

* Section: Workday Web Services- Details the Get_Job_Profiles operation and its XML output structure, including

`<wd:Job_Profile_Data>` and `<wd:Job_Code>`.

* Section: XPath Syntax- Explains how to navigate XML hierarchies in Workday XSLT, using relative paths like

`wd:Job_Profile_Data/wd:Job_Code` from a `<wd:Job_Profile>` context.

* Workday Community SOAP API Reference - Provides examples of XPath navigation for Workday web service responses.

Option C is the verified answer, as it correctly selects the `<wd:Job_Code>` value using the appropriate XPath syntax within the `<wd:Job_Profile>` template context.

質問 # 53

How does an XSLT processor identify the specific nodes in an XML document to which a particular transformation rule should be applied?

- A. Named templates explicitly call processing for designated elements.
- B. The processor targets nodes based on declared namespace prefixes.
- **C. The processor matches nodes using XPath expressions within templates.**
- D. The stylesheet element directs the processor to specific XML sections.

正解: C

解説:

In XSLT, the processor applies transformation rules by matching nodes using XPath expressions inside `<xsl:template match=''/>` statements.

"Templates define the rule, and XPath expressions determine which nodes they apply to." This is the foundational mechanism by which XSLT processes XML data.

Why the others are incorrect:

B . The `<xsl:stylesheet>` element defines scope, not node matching.

C . `<xsl:call-template>` invokes a named template but does not itself match nodes.

D . Namespace prefixes are used within XPath, but node matching is based on XPath.

質問 # 54

What is a key function and primary benefit of using a Document Transformation Connector within the integration capabilities of Workday?

- A. It provides functionality for defining a business process to manage both the connector integrations and document transformations output files.
- B. It serves as the principal tool for establishing and maintaining secure connections of connector integrations with various external systems.
- **C. It plays a crucial role in converting the XML outputs generated by connector integrations into diverse formats and allows for data formatting and validation.**

- D. It enables the application of intricate calculations on Workday data before it is extracted by other integration tools for external transmission.

正解: C

解説:

The Document Transformation Connector is used in Workday to process and reformat XML outputs - often from Core Connector or EIB integrations - into custom formats like CSV, JSON, or flattened XML.

"The primary role of the Document Transformation Connector is to apply XSLT-based formatting, data reorganization, and validation to the output of Workday integrations before delivery to downstream systems." This is especially useful when third-party vendors require a specific format not natively supported by the integration system.

Why the other options are incorrect:

- * A. Managing business processes is not a DT Connector's function.
- * B. Calculations are not the main purpose - that's more for Calculated Fields or Studio.
- * D. While security is essential, secure connections are managed through Workday's integration system and transport configuration, not the DT connector.

Reference: Workday Pro: Document Transformation Overview - Use Cases for Document Transformation Connector
Workday Integration Community: Best Practices for XSLT and Output Formatting

質問 # 55

What attribute(s) can go into the `xsl:stylesheet` element?

- A. XSLT Version & Namespaces
- B. XSLT Version & Encoding
- C. XML Version & Namespaces
- D. Namespaces & Encoding

正解: A

解説:

The `<xsl:stylesheet>` element is the root element in an XSLT document. It must include:

- * XSLT Version - This defines the XSLT specification version being used (e.g., `version="1.0"` or `version="2.0"`).
- * Namespaces - XSLT operates within an XML namespace (`xmlns:xsl="http://www.w3.org/1999/XSL/Transform"`), which is required to define the transformation rules.

Breakdown of Answer Choices:

- * A. XSLT Version & Namespaces # (Correct)
- * The `<xsl:stylesheet>` element requires both the XSLT version and the namespace declaration for proper execution.
- * Example:

xml

CopyEdit

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

- * B. XSLT Version & Encoding # (Incorrect)
- * Encoding (`encoding="UTF-8"`) is a property of the XML declaration (`<?xml version="1.0" encoding="UTF-8"?>`), not an attribute of `<xsl:stylesheet>`.
- * C. XML Version & Namespaces # (Incorrect)
- * XML version (`<?xml version="1.0"?>`) is part of the XML prolog, not an attribute of `<xsl:stylesheet>`.
- * D. Namespaces & Encoding # (Incorrect)
- * Encoding is not an attribute of `<xsl:stylesheet>`.

Final Correct Syntax:

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

This ensures that the XSLT file is processed correctly.

Workday Pro Integrations Study Guide References:

- * ReportWriterTraining.pdf - Chapter 9: Working With XML and XSLT covers XSLT basics, including the required attributes for `<xsl:stylesheet>`.
- * Workday_Advanced_Business_Process_part_2.pdf - Chapter 5: Web Services and Integrations details how Workday uses XSLT for transformations.

質問 # 56

あなたはその他のWorkday Workday-Pro-Integrations「Workday Pro Integrations Certification Exam」認証試験に関するツールサイトでも見るかも知れませんが、弊社はIT業界の中で重要な地位があって、Topexamの問題集は君に100%で合格させることと君のキャリアに変らせることだけでなく一年間中で無料でサービスを提供することもできます。

[illegible]

2026年Topexamの最新Workday-Pro-Integrations PDFダンプ および Workday-Pro-Integrations試験エンジンの無料共有: <https://drive.google.com/open?id=1qNfZvxjTlazNpv1Wzm0R4O8Aqa5ekmFo>