

DUMPSBOSS.

Talend Data Integration Certified Developer Exam

Talend Data-Integration-Developer

Version Demo

Total Demo Questions: 7

Total Premium Questions: 76

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

Which section in Talend Studio allows you to graphically connect components in a Job to run a dataflow process?

- A. Design workspace
- B. Component view
- C. Repository
- D. Code

ANSWER: A

Explanation:

Design workspace is correct because it is the central graphical canvas used to build a Talend Job. In this workspace, a developer drops components onto the Job design area, configures their properties, and creates links between them to define how records move through the process. These graphical connections represent the Job's execution and dataflow logic, such as a row flow from an input component to a transformation component and then to an output component. The workspace also supports the creation of trigger links, which control execution order, and reject or lookup flows where applicable. As the Job is designed, Talend Studio generates the corresponding Java code behind the scenes, but the developer normally creates and maintains the integration process visually in the Design workspace rather than by manually writing that code. This visual design model is a core feature of Talend Studio: components encapsulate integration operations, while their links define the relationships and flow between those operations. Talend documentation describes Job design as assembling components and connecting them in the Studio design area to implement an integration flow. See the [Talend Studio User Guide: Designing a Job](#) and the [Talend Studio user interface documentation](#).

QUESTION NO: 2

You are building a Job that consists of two subJobs being sequentially executed. The first subJob completes successfully.

Which component in the first subJob triggers the execution of the second subJob?

- A. Start component
- B. Final component
- C. Input component
- D. Any component

ANSWER: D

Explanation:

Any component is correct because sequential execution between Talend subJobs is controlled by a trigger connection, specifically an `OnSubjobOk` trigger, rather than by a dedicated Start, Final, or input component. In Talend Studio, you can create an `OnSubjobOk` connection from a component in the first subJob to the starting component of the next subJob. Talend evaluates this trigger when the subJob containing the source component has completed successfully. The connected second subJob is then launched.

This is an important distinction in Job orchestration: `OnSubjobOk` represents successful completion at the subJob level, so it is appropriate when the next processing stage must not begin until all work in the preceding subJob is complete. The component used as the source is therefore a design choice; it does not need to be a special component type. Configure the trigger according to the desired dependency and connect it to the component that begins the subsequent processing flow. Talend's Studio documentation describes trigger connections and their role in controlling Job execution order: [Trigger connections](#). See also the official [Talend Studio User Guide](#).

QUESTION NO: 3

In Talend Studio, you have access to a references project named `project_ref`. You need to reuse a Job named `testJob` from the `project_ref` in your main Job.

- A. Right-click Job Designs and click import items.
- B. Navigate to Metadata, @ project ref, Job Designs, Standard, then select `testJob`.
- C. Right-click Job Designs, click Create Standard Job, and enter `testJob` as the Job name.
- D. Navigate to @ Referenced project, @ project_ref, Job Designs, Standard, then select `testJob`.

ANSWER: D

Explanation:

Navigate to @ Referenced project, @ project_ref, Job Designs, Standard, then select `testJob`. In Talend Studio, a referenced project makes repository items from another project available to the current project without copying those items into the current project. After the reference has been configured in the current project's Project Settings, Talend displays a *Referenced project* node in the Repository tree. Expanding that node and then the named project exposes its repository structure, including Job Designs and the Standard folder containing the reusable Job.

Selecting or dragging *testJob* from that location into the main Job design creates a reusable Job reference, typically represented through the appropriate Job component configuration. This approach preserves the source Job in *project_ref* as the maintained artifact and lets the main Job invoke it rather than creating a duplicate. Referenced-project content is accessed in read-only mode, which is important for controlled reuse of shared Jobs, routines, metadata, and other repository artifacts across projects. Talend's Studio documentation describes referenced projects and their Repository-tree access model in its [Referenced projects documentation](#). Additional guidance on organizing and using repository artifacts is available in the [Talend Studio Repository documentation](#).

QUESTION NO: 4

Which statement is true about the Sync columns button on the Basic settings tab of the Component view? Choose 2 answers

- A. Retrieves and synchronizes the output file schema with the input file schema for tFileOutputed limited.
- B. Retrieves the schema of the current component to match the Next component
- C. Retrieves the schema from the input component connected in the lob.
- D. Retrieves and synchronizes the output schema with the input schema for tJavaRow.

ANSWER: C D

Explanation:

The Sync columns button updates a component's schema from the schema supplied by its incoming connection. Therefore, it retrieves the schema from the input component connected in the Job, allowing the receiving component to inherit newly added, removed, or modified input columns. This is particularly useful after changing an upstream component, because it keeps downstream column metadata aligned without manually editing each column definition.

For tJavaRow, which has both an input flow and an output flow, synchronizing columns aligns the component's output schema with its input schema. This supports the normal tJavaRow use case: Java code can transform values while the component retains a corresponding row structure for the outgoing connection. Talend Studio documents schema propagation and synchronization as part of managing component schemas and metadata in Jobs. See the [Talend Studio documentation on propagating schema changes](#) and the [tJavaRow standard properties reference](#).

QUESTION NO: 5

Where can you specify the remote JobServer to execute a Job?

- A. Job settings section of the Project Settings windows
- B. Extra tab in the Job view
- C. Run/Debug section in the Preference window
- D. Target Exec tab in the Run view

ANSWER: D

Explanation:

The **Target Exec tab in the Run view** is where Talend Studio defines the execution target for the currently open Job. This tab lets a developer choose between local execution in Studio and remote execution through a configured JobServer. When remote execution is selected, the developer can select the JobServer connection that should receive and run the generated Job artifact. This makes the setting specific to how that Job is launched rather than a general Studio preference or a Job-design feature.

A Talend JobServer is the runtime service used to execute Jobs on a remote machine. Selecting it from the Target Exec configuration enables Studio to submit the Job for that remote runtime environment, where the JobServer handles execution and returns relevant status information. The Run view is therefore the appropriate place because it centralizes launch-time settings, including the target environment, context, and execution parameters. Talend documentation describes the Run view as the Studio area for configuring and launching Jobs, while JobServer documentation covers its role as a remote execution service. See the [Talend Studio Run view documentation](#) and the [Talend JobServer documentation](#).

QUESTION NO: 6

Which statements are true about Use or register a shared DB Connection option in Talend Studio?

Choose 3 answers

- A. Compatible with the Use an independent process to run subJob option of tKunJob component
- B. Compatible to be snared between parent and child Jobs
- C. Compatible with The Use dynamic job option of TRunJob component
- D. Incompatible with the Use an independent process to run subJob option of tRunJob component
- E. Incompatible with the Use dynamic job option of tRunJob component

ANSWER: B D E

Explanation:

“Compatible to be snared between parent and child Jobs” is correct because Talend can reuse a database connection established by a parent Job in a child Job invoked through tRunJob. The parent registers the connection and the child accesses that shared connection, avoiding the overhead of creating a separate physical database connection. This capability requires both Jobs to execute within the same JVM process.

“Incompatible with the Use an independent process to run subJob option of tRunJob component” is correct because an independently executed child Job runs in another JVM. A shared database connection is an in-memory Java object, so it cannot be passed directly across JVM process boundaries.

“Incompatible with the Use dynamic job option of tRunJob component” is also correct. Dynamic Job execution resolves the called Job at runtime rather than using a statically defined child Job, and Talend does not support passing the shared connection metadata required for this reuse pattern. Talend documents shared DB connections as a mechanism for connection reuse between eligible components and parent/child Jobs, with the relevant tRunJob execution constraints. See [Talend documentation: Use or register a shared DB Connection](#) and [Talend documentation: tRunJob standard properties](#).

QUESTION NO: 7

A file-processing Job must invoke a downstream subjob once for every input row. The downstream subjob needs to use values from the current row, such as the file name and processing date.

Which statements are true about using tFlowTolterate?

Choose 2 answers

- A.** It converts each incoming row into an iteration.
- B.** It merges all incoming rows into one output row.
- C.** It makes values from the current row available through globalMap.
- D.** It sends the input schema unchanged through a Row connection.

ANSWER: A C

Explanation:

tFlowTolterate converts each incoming row into an iteration, so the following subjob connected through an Iterate link can run once per row. It also makes values from the current input row available through the Job's globalMap for use by components in the iterated subjob.

tFlowTolterate does not produce an ordinary outgoing row flow, and it does not combine all incoming rows into one iteration. It is used when row data must drive iterative control flow rather than be passed directly through a Row connection.