PassQueen
a.k.a dumpsqueen.com

WSO2 Certified Enterprise Integrator 6 Developer (6.5 version)

WSO2 Enterprise-Integrator-6-Developer

Version Demo

Total Demo Questions: 13

Total Premium Questions: 139

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

Which mediator should be used when a sequence must invoke a back-end service and continue processing the returned response in the same mediation flow?

A. Call mediator

B. Send mediator

C. Respond mediator

D. Drop mediator

ANSWER: A

Explanation:

Call mediator is correct because it invokes an endpoint and resumes the mediation sequence after the response is received. This enables subsequent mediators to inspect, transform, enrich, or route the response payload. The Call mediator supports non-blocking invocation by default and can also be configured for blocking invocation where that is required. In contrast, the Send mediator normally hands the message to an endpoint and ends the current sequence; response processing is then handled through the configured out sequence or fault sequence. The Callout mediator is primarily intended for synchronous blocking SOAP service invocations and is not the general-purpose choice for this scenario. See the Apache Synapse Call mediator reference and the Apache Synapse Send mediator reference.

QUESTION NO: 2

Which statement is TRUE about Java Message Service (JMS) support in the ESB profile?

A. The ESB profile can be configured as both a JMS Producer and Consumer.

B. JMSsupports point-to-point communication models only.

C. The ESB profile can be configured as a shared topic listener.

D. Both statements a and c are correct.

ANSWER: D

Explanation:

The combined statement that the ESB profile can act as both a JMS producer and consumer and can be configured as a shared topic listener is correct. In WSO2 Enterprise Integrator 6, the JMS transport supports sending messages from mediation flows to a JMS broker through the JMS sender configuration; this is the producer role. It also supports receiving messages from JMS destinations through the JMS listener configuration, after which messages are processed by the ESB mediation sequence; this is the consumer role. JMS itself supports both queue-based point-to-point messaging and topic-based publish-subscribe messaging. For topic consumption, WSO2's JMS listener configuration provides support for shared subscriptions/listeners, allowing multiple consumers to share the work associated with messages published to a topic where the selected JMS provider supports that capability. Therefore, both capabilities included in the combined statement are supported by the ESB profile's JMS transport. WSO2's JMS transport documentation describes configuring JMS senders and listeners, including destination and subscription-related properties: Working with JMS Transports. The JMS API's queue and topic messaging models are also defined by the Jakarta Messaging specification.

QUESTION NO: 3

Identify the INCORRECT statement with respect to endpointtimeout:

A. For all endpoint timeout values. Endpoint Timeout >= http.socket.timeoutThe following is a correct endpoint definition with a timeout value and a response action: url="http://localhost:8281/services/SimpleStockQuoteService" >

B. 120000fault

C. It is possible to define a responseAction to be taken in case of an endpoint timeout within the endpoint definition.

D. Endpoint timeout specifies the timeout duration for an endpoint and allows different timeouts to be specified for different endpoint definitions.

ANSWER: A

Explanation:

The statement “For all endpoint timeout values. Endpoint Timeout >= http.socket.timeout” is incorrect. In WSO2 Enterprise Integrator/Synapse, an endpoint timeout is the endpoint-level period used to determine that a backend has not responded in time and to apply the configured timeout behavior. It must be configured to expire before the underlying HTTP socket timeout. Therefore, the endpoint timeout duration should be less than the `http.socket.timeout` value configured for the transport. This ordering ensures that Synapse can identify the endpoint timeout and execute its configured response action, such as raising a fault, rather than having the lower-level socket timeout occur first. Endpoint timeout configuration is defined within an address endpoint’s `<timeout>` element, which can contain a duration and a response action. This gives each endpoint its own timeout policy while preserving the transport-level socket timeout as the upper limit. See the WSO2 documentation on Endpoint Timeouts and the Apache Synapse configuration language reference.

QUESTION NO: 4

Which of the following is NOT an action to take in order to tune a new ESB profile server for high performance?

A. Synapse configuration level settings

B. JVM-level settings

C. WS02 Carbon platform level settings

D. OS-level settings

ANSWER: A

Explanation:

Synapse configuration level settings is the correct choice because tuning a newly deployed ESB profile for baseline high performance is primarily a server-runtime exercise, rather than an adjustment to a specific Synapse mediation configuration. Initial tuning focuses on allocating and managing JVM resources effectively, configuring the Carbon runtime’s thread pools and caches, and ensuring that the operating system provides sufficient file descriptors, network capacity, and process resources. These settings establish the throughput and concurrency characteristics of the server before it runs production integrations.

Synapse configurations define the mediation logic for individual services and flows. Although a particular integration can later be optimized—for example, by reducing unnecessary mediation, payload construction, or blocking calls—that is application-specific optimization rather than a standard platform-level action for preparing a new ESB profile server. WSO2’s performance-tuning guidance treats the runtime environment and platform configuration as the main areas to address when establishing an efficient server baseline. See the WSO2 EI documentation on performance tuning and the Apache Synapse reference for the role of Synapse configuration.

QUESTION NO: 5

Which statement about large file transfer using ESB profile is true?

A. To enable binary file transfer, you must enable the message builder and formatter for the relevant content type in axis2.xml.

B. You can enable streaming within your ESB profile configuration using a parameter.

C. All of the above.

D. Files can be transferred as binary files using streaming to reduce processing and memory requirements.

ANSWER: C

Explanation:

“All of the above.” is correct because ESB Profile large-file transfers can use binary streaming, which avoids constructing the complete file payload in memory and therefore reduces processing overhead and memory consumption. This is particularly important when a proxy service receives a file through the VFS transport and sends it onward to another file location or endpoint. Streaming is enabled through the relevant VFS streaming parameter in the ESB configuration, allowing the file content to be handled as a stream rather than as a fully built message. Binary content handling also requires the applicable Axis2 message builder and message formatter configuration for the content type, such as `application/octet-stream`. The builder enables the incoming binary payload to be interpreted correctly, while the formatter enables it to be written correctly on the outbound side. Together, these settings allow the ESB Profile to transfer large files efficiently without unnecessary payload transformation or in-memory buffering. WSO2 documents the VFS transport and its file-processing behavior in the VFS Transport documentation; its guidance for this use case is available in Transferring Large Files.

QUESTION NO: 6

In the proxy service, it is necessary to call one of the several back-end services depending on the format of the message. Forinstance, the incoming message to the proxy service might have a property called "company". If company == 'IBM', we need to call the IBM service, and if company == 'Google', we need to call the Google service. To implement this scenario, what would be thebest option available in the ESB profile?

A. None of these approaches would implement this scenario.

B. Use the Switch mediator.

C. Write a custom Class mediator and implement the custom routing logic inside that mediator.

D. Use the Script mediator and implement the above routing logic.

ANSWER: B

Explanation:

Use the Switch mediator. It is designed for content-based routing in WSO2 Enterprise Integrator: it evaluates an XPath or JSONPath expression and selects a matching case based on a regular expression. In this scenario, the mediator can read the `company` property through an expression such as `get-property('company')`. A case matching `IBM` can contain the endpoint invocation for the IBM back-end service, while a case matching `Google` can invoke the Google back-end service. Each case may include the required mediators, such as a Call mediator or Send mediator, to dispatch the message to its selected endpoint.

This keeps routing declarative and within the proxy-service mediation sequence. It also makes the routing rules easy to extend: additional company values can be added as cases, and a default branch can define the behavior for an unrecognized or missing value. The Switch mediator is therefore the appropriate ESB-profile mechanism where the target back-end service is selected according to a message property. WSO2 documents the mediator’s source-expression and case-regex model in the Switch mediator reference. The related routing pattern is also described in WSO2’s content-based routing example.

QUESTION NO: 7

In which folder should you put custom mediator in the EI distribution?

A. If your Class mediator is a regular non-OSGI JAR file, copy it into the /repository/classmediators directory.

B. Both a and c are correct.

C. If you develop your mediator as a OSGI bundle, copy jar file into /dropins directory.

D. If your mediator is a regular non-OSGI JAR file, copy it into the \lib directory.

ANSWER: D

Explanation:

If your mediator is a regular non-OSGI JAR file, copy it into the \lib directory. is correct for the EI 6.5 distribution. The `<EI_HOME>/lib` directory is the deployment location used to make an ordinary custom-mediator JAR available to the Enterprise Integrator runtime. On startup, EI includes compatible libraries from this location in its runtime classpath, allowing a Class Mediator configuration to load the class specified by its fully qualified name and invoke its mediation logic. The JAR should contain the mediator implementation and all required dependencies, or those dependencies must otherwise be available to the runtime. After copying the library, restart the server so that its class loading and mediation configuration are initialized with the custom implementation. This approach is appropriate when the mediator is packaged as a conventional JAR rather than being deployed through a separate OSGi-bundle lifecycle. WSO2's Enterprise Integrator documentation describes custom mediator development and deployment conventions in its Creating a Custom Mediator guide. The broader EI 6.5 documentation is available from the WSO2 Enterprise Integrator 6.5 documentation portal.

QUESTION NO: 8

Identify the correct statement with respect to mutualSSL.

A. Mutual SSL provides message level security.

B. Mutual SSL provides higher performance.

C. All of the above are correct.

D. Both client and server have to authenticate each other and require certificates to be configured at both server and client ends.

ANSWER: D

Explanation:

Both client and server have to authenticate each other and require certificates to be configured at both server and client ends. Mutual SSL, also called mutual TLS or two-way SSL, extends the normal TLS handshake by requiring the client to present an X.509 certificate as well as the server. The server proves its identity to the client using its server certificate, and the client proves its identity to the server using its client certificate. Each side must be able to validate the certificate presented by its peer, which means the relevant trusted certificate or issuing certificate authority must be available in the corresponding trust configuration. In WSO2 Enterprise Integrator deployments, this commonly involves configuring key and trust stores so that the listener can validate client certificates and the outbound client can validate the remote server certificate. This establishes authenticated transport connections and is especially useful where service access must be restricted to known calling systems rather than authenticated only by application-level credentials. WSO2's transport-security guidance describes configuring SSL/TLS materials through keystores and truststores, while the TLS specification defines certificate-based peer authentication during the handshake. See WSO2 keystore and truststore documentation and RFC 5246 certificate request and client authentication.

QUESTION NO: 9

If you are running two or more ESB profile servers on the same physical machine, it is necessary to offset ports. Select the correct statement that describes how to offset ports in the ESB profile.

A. Go to /conf/axis2, open the axis2.xml file, and add the following top-level configuration:

B. Both statement 1 and 2 are correct.

C. Both statement 1 and 2 are incorrect.

D. Go to /conf, open the carbon.xml file, locate the following configuration and increase the Offset value: 0.

ANSWER: D

Explanation:

Go to `<ei_home>/conf`, open `carbon.xml`, and increase the value of `<Ports><Offset>...</Offset></Ports>` is correct because the Carbon port offset is the supported server-wide mechanism for operating multiple WSO2 Enterprise Integrator ESB profile instances on one host. The offset is an integer added to each default listening port of that EI runtime. For example, when one instance uses the default offset of `0`, setting a second instance to `1` changes its default ports by one, preventing conflicts while retaining the normal port configuration model. The offset must be set before starting the additional server instance, and each simultaneously running instance needs a distinct offset value. This setting belongs in the Carbon configuration because EI 6.5 is built on the WSO2 Carbon platform, which centrally calculates the effective transport and management ports using this value. WSO2's EI documentation describes changing the port offset for multiple product instances in its default-port configuration guidance: Changing Default Ports. The EI 6.5 distribution's Carbon configuration also contains the `Ports` and `Offset` setting: EI 6.5 carbon.xml.

QUESTION NO: 10

Consider thefollowing scenario: you are working on an ESB integration project, and you want to work with messages that contain PDF files as the payload. You also need to filter some messages based on the content of the PDF files. Identify the step you would need to take to fulfill this requirement.

A. Implement a custom message formatter to work with PDF payloads and register it in the /conf /axis2/axis2.xml file under the section.

B. Implement a custom message builder to work with PDF payloads and register it in the /conf /axis2/axis2.xml file under the section.

C. Both steps a and b are required.

D. WS02 ESB can handle PDF payloads by default, so there is no need to build a custom message builder/formatter.

ANSWER: B

Explanation:

Implement a custom message builder to work with PDF payloads and register it in the <ei_home>/conf /axis2/axis2.xml file under the <messagebuilder> section. is correct because a message builder is invoked when the Enterprise Integrator receives an incoming message. Its role is to interpret the incoming payload according to its content type and construct the message representation that mediators can process. A PDF is binary data and does not provide an XML object model that can be directly evaluated by content-aware mediators. Therefore, a custom builder must extract or transform the relevant PDF content into a form available to the mediation flow, such as text or XML, enabling filtering based on that content. The builder is registered in the Axis2 configuration and associated with the relevant PDF MIME type so that it is selected for inbound PDF messages. This follows the Axis2 message-processing model, in which message builders create the internal representation of incoming data before mediation logic accesses it. WSO2's guidance on binary payload handling and Axis2's message-builder documentation provide the applicable configuration and processing concepts: WSO2 EI binary data documentation and Apache Axis2 Message Builders and Formatters.

QUESTION NO: 11

Which statement is TRUE about proxy services?

A. Proxy services should be implemented as pure Java classes.

B. Only HTTP and HTTPS transports can be used to receive and send messages from the proxy services.

C. Proxy services allow you to perform necessary transformations and introduce additional functionality without changing your existing service.

D. A proxy service is not externally visible and can't be accessed using a URL similar to a normal web service address.

ANSWER: C

Explanation:

Proxy services allow you to perform necessary transformations and introduce additional functionality without changing your existing service. In WSO2 Enterprise Integrator, a proxy service provides a virtual service endpoint in front of an existing service. Messages received by the proxy can be processed through mediation logic before they are forwarded to a backend endpoint, and responses can likewise be mediated before being returned to the caller. This enables integration concerns such as message transformation, validation, routing, security enforcement, logging, throttling, and protocol bridging to be added independently of the implementation of the original service. As a result, consumers can use the proxy-facing service while the underlying service remains unchanged, which is particularly valuable when integrating legacy services or when applying cross-cutting policies consistently. A proxy service is configured as an EI/Synapse artifact with defined transports, a sequence or mediation flow, and optionally a target endpoint; it does not require modifying the backend application's source code. WSO2's proxy-service documentation describes proxy services as virtual services that mediate requests and responses and can route messages to other services. See WSO2 EI 6.5: Creating a Proxy Service and WSO2 documentation: Proxy Services.

QUESTION NO: 12

Which statement is TRUE about throttling in the ESB profile?

A. All of the above are correct.

B. Throttling isonly supported at the mediation level.

C. The throttle mechanism is used to control access to the services at different levels.

D. Mediation level throttling works in distributed fashion only when ESB nodes are configured as a stateful cluster which is a membership based cluster.

ANSWER: C

Explanation:

The throttle mechanism is used to control access to the services at different levels. In WSO2 Enterprise Integrator's ESB profile, throttling regulates the flow of requests so that a backend service or integration flow is protected from excessive traffic and can continue operating within defined capacity limits. The Throttle mediator supports policies that limit access based on criteria such as an IP address, domain, tenant, or an application-specific identifier carried in the message. A policy can define limits such as the number of requests permitted during a specified time unit, and requests that exceed the limit can be rejected or handled through a configured response path. This makes throttling useful both for protecting service availability and for enforcing consumption limits across integration endpoints. Throttling is therefore not limited to one narrow point in an integration; it is a service-access control capability that can be applied at relevant levels of the message-processing architecture. The WSO2 documentation for the Throttle mediator describes its role in restricting message traffic using configured policies, while Apache Synapse documents the underlying mediator behavior and policy-based access control model. See the WSO2 Throttle Mediator documentation and the Apache Synapse Throttle mediator reference.

QUESTION NO: 13

Which statement about the Entitlement mediator is true?

A. Entitlement mediator intercepts requests and evaluates the actions performed by the user against an extensible Access Control MarkupLanguage (XACML) policy.

B. All of the above.

C. When configuring the mediator, you specify which sequence to execute depending on the Entitlement mediator result.

D. Entitlement mediator can only work with WS02 Identity Server as the PolicyDecision Point.

ANSWER: A

Explanation:

The statement “Entitlement mediator intercepts requests and evaluates the actions performed by the user against an extensible Access Control MarkupLanguage (XACML) policy.” is correct. In WSO2 Enterprise Integrator, the Entitlement mediator is used to enforce authorization decisions in an integration flow. It obtains the relevant request context, such as the subject, resource, action, and environment attributes, and submits an entitlement request to a Policy Decision Point. The request is assessed against XACML policies, which express fine-grained authorization rules independently of the application or service implementation. Based on the resulting permit, deny, indeterminate, or not-applicable decision, the mediation flow can enforce the authorization outcome before the protected service is invoked. This approach separates policy administration and policy decision logic from the service integration logic, allowing access-control rules to be centrally managed and consistently applied. XACML defines the standard model for requesting and evaluating attribute-based authorization decisions; see the OASIS XACML 3.0 Core Specification. WSO2’s Enterprise Integrator documentation describes the mediator and its role in entitlement-based authorization at Entitlement Mediator.