MQ Input Node

This node is used to read or browse a message from a queue which is present on a queue manager.

Some Important Properties of MQ Input Node

Queue Name

Here we provide the queue name from which MQ input node will read the message for further processing.

Input Message Parsing

This property defines which type of message (Domain) your MQ Input Node is going to process. It has multiple options to select. Mostly we work with DFDL, XMLNSC, JSON and BLOB. In this course, we will be dealing with all these message domains.

Parse Timing

On Demand: In this case, validation/parser errors will not be detected. Message flow will not throw any error. Message will be successfully parsed till the point it is correct.

Complete: This value is of most importance among all three values. In this case, message will be completely parsed, and all the validation/parser error will be detected, and message flow will throw an exception if there is a problem.

Immediate: This value is of least importance. It is same as Complete parsing; the only difference comes in case of MRM domain. If domain is MRM, then everything in message is parsed and validated except those complex types with a Composition of Choice or Message that cannot be resolved at the time.

Opaque Elements

This is rarely used, but it is very important to know about this property as mostly people face question related to this in interviews.

This facility is provided by XMLNSC domain.

If you have a very big input message and this message contains elements which are never used or referenced in your message flow then you can define opaque parsing for these elements meaning that logical tree for these elements will not be created which will reduce cost of parsing and also will improve the performance as creating logical tree for a very big message may hamper the performance.

Transaction Mode

It defines how my message flow is going to process the message and how my transaction is going to be handled.

Yes: If transaction mode on MQ input node is ‘Yes’, then in case of any unhandled exception, my original message which was received by MQ input node will go to Backout queue. If there is no backout queue, then it will go to dead letter queue. If there is no dead letter queue as well, then it will continuously try to search for the place to reside and in this process it may go inside an infinite loop which may create some headache and execution group may show some unwanted behavior.

No: If transaction mode on MQ input node is ‘No’, then in case of any unhandled exception, my original message which was received by MQ input node will be simply lost. We cannot find this message anywhere.

Automatic: If my transaction mode is ‘Automatic’, then behavior will be governed by persistence nature of the queue/message. If my queue/message is persistent, then behavior will be same as if my transaction mode is

'YES'. On the other hand, if my queue/message is not persistent, then behavior will be same as if my transaction mode is 'No'.

Validation

Validate

None: Default value. It is used when no validation is required.

Content: Used for content validation. For example, checking the composition.

Content and Value: Used for validating the value along with the content. For example, suppose some value is defined as an integer. So, it will validate if it is integer or not.

Note: For message domains like SOAP, DFDL and XMLNSC, always content and value check will be performed even if we select content only.

Failure Action:

Exception: It gives only first validation failure.

Exception List: It gives all the validation failures.

Local Error Log: Message flow will not show any failure in this case. Error can be seen in event logs.

User Trace: Message flow will not show any failure in this case. Error can be seen in user trace file.

MQ Output Node

MQ output node is used to write a message on a queue which is defined on a queue manager.

Some Important Properties of MQ Input Node

Queue Name

Here we provide the queue name on which MQ output node will write the message.

Destination Mode

Queue Name: If we select this property, the message will be written on the queue which is defined on Queue Name property.

Reply To Queue: If we select this property, the message will be written on the ReplyToQ field in MQMD header.

Destination List: if we select this property, the message will be written on list of queues provided in local environment tree.

Transaction Mode

Yes: If transaction mode on MQ output node is yes, then whatever operation we are doing will be committed only if my transaction is complete. In case of any exception, my transaction will be rolled back i.e. operations will not be committed and everything will be in original state.

No: If transaction mode on MQ output node is No, then whatever operation we are doing will be committed doesn't matter if my transaction is successful or some exception occurred in between.

Automatic: In case of automatic, it will inherit the transaction mode property from input node.