[← Implementing security authentication using LDAP in IBM Integration Bus](https://eaideveloper.wordpress.com/2014/01/29/implementing-security-authentication-using-ldap-in-ibm-integration-bus/)

[Integrate IBM Integration Bus with IBM Operational Decision Management using Decision Service Node →](https://eaideveloper.wordpress.com/2014/03/02/integrate-iib-with-ibm-odm/)

**[Environment and Local Environment trees in IBM Integration Bus](https://eaideveloper.wordpress.com/2014/02/09/environment-and-local-environment-trees-in-iib/)**

Posted on [February 9, 2014](https://eaideveloper.wordpress.com/2014/02/09/environment-and-local-environment-trees-in-iib/)by [vishnu](https://eaideveloper.wordpress.com/author/vishnurajnr/" \o "View all posts by vishnu)

**Environment and Local Environment trees in  IBM Integration Bus**

Yes, this topic is been discussed many times in many places. And the InfoCenter is been improved a lot in addressing this topic. Yet, I just want to put some simple analysis on one of the most used tree structures in message flow development – Local Environment and Environment trees.

**Local Environment** and **Environment** are message trees used by the message flow during a message processing. The Lifetime of this tree structure may be throughout the flow or limited some nodes. Use these trees to carry variables that need to be accessed in nodes throughout the flow.

Both trees are part of logical message tree which we can use to **store data during the message flow processing.**Local Environment has some special use such as to set the message destinations. More details are following.

**Environment** variable will live the whole thread lifetime of a flow. **LocalEnvironment**tree may or may not live throughout the thread lifetime. That means there is a mechanism which decides whether or not to pass the LocalEnvironment trees throughout the flow. ’Compute mode’ property of a compute node will decide whether or not the local environment tree is copied to the downstream nodes of the flow.

Let’s analyse this compute mode and how it will affect the Local Environment tree structure.

When a message comes enter into a compute node, a new Local Environment tree will be created which is visible as ‘Output Local Environment’.

|  |  |
| --- | --- |
| Compute Mode | OutPut Tree |
| Message | New **OutputRoot** + InputlocalEnvironment + InputExceptionList |
| LocalEnvironment | InputRoot + new **OutputLocalEnvironment**+ InputExceptionList |
| LocalEnvironment and Message | New **OutputRoot** + new **OutputLocalEnvironment** + InputExceptionList |
| Exception | InputRoot + InputLocalEnvironment + new **OutputExceptionList** |
| Exception and Message | New **OutputRoot** + InputLocalEnvironment + new **OutputExceptionList** |
| Exception and LocalEnvironment | InputRoot + new **OutputLocalEnvironment**  + new **OutputExceptionList** |
| All | **New Output Tree + New Output Local Environment + new Output ExceptionList** |

Eg: Your compute mode is ‘Message’ and in the compute node you have copied the inputLocalEnvironment tree to OutputLocalEnvironment and added an element in the OutputLocalEnvironment tree. But we cannot get the changes after the compute node since it will not be propagated.

The above behaviour is not applicable for Environment tree since it can propagate to upcoming nodes regardless of the compute mode property. If you want to create your own information, create it in the environment tree in a sub tree called Variables.

The environment tree differs from the local environment tree in that a single instance of it is maintained throughout the message flow.

LocalEnvironment has several predefined children such as **LocalEnvironment.Variables, LocalEnvironment.Destination**etc.

**Imp Note:** The input message trees are not passed to the output unless they are copied explicitly from the Input to the Output.

Some of the most commonly used destination trees are listed below:

|  |  |
| --- | --- |
| Node | Tree |
| MQOutPut | LocalEnvironment.Destination.MQ.XXX |
| FileOuput | LocalEnvironment.Destination.File.XXX |
| EmailOutPut | LocalEnvironment.Destination.Email.XXX |
| FTE Ouptput | LocalEnvironment.Destination.FTE.XXX |
| HTTP Request | LocalEnvironment.Destination.HTTP.XXX |
| JMSOutput | LocalEnvironment.Destination.JMSDestinationList.XXX |
| SOAP Reply, SOAP request, SOAP AsyncRequest | LocalEnvironment.Destination.SOAP.XXX |
| TCPIP Output, TCPIP Server | LocalEnvironment.Destination.TCPIP.XXX |
| CICSRequest | LocalEnvironment.Destination.CICS.XXX |
| CORBARequest | LocalEnvironment.Destination.CORBA.XXX |

Some of the most commonly used localenvironment trees generated by the \**input* nodes are listed below:

|  |  |  |
| --- | --- | --- |
| **Related Node** | **Tree** | **Usage** |
| FileInput | LocalEnvironment.***Wildcard*** | File Name pattern |
| MQ Output | LocalEnvironment.WrittenDestination | transport-specific information |
| *Adapter Nodes* | LocalEnvironment.Adapter.*XXX* | Information stored by Adapter nodes |
| CDInput | LocalEnvironment.CD.*XXX* | Info stored by the CDInput node |
| DatabaseInput | LocalEnvironment.Database.XXX | Info propagated from the node |
| DecisionService  node | LocalEnvironment.DecisionServices |
| FileInput node | LocalEnvironment.File |
| FTEInput node | LocalEnvironment.FTE.XXX |
| JMSReceive node | LocalEnvironment.JMS |
| Mapping node | LocalEnvironment.Mapping | override the mapping routine |
| EndpointLookup, RegistryLookup | LocalEnvironment.ServiceRegistry |  |
| SOAPInput, SOAPAsyncResponse, SOAPRequest | LocalEnvironment.SOAP |  |
| TCPIPClientReceive, TCPIPServerReceive | LocalEnvironment.TCPIP |  |

[Infocenter](http://pic.dhe.ibm.com/infocenter/wmbhelp/v9r0m0/index.jsp?topic=%2Fcom.ibm.etools.msgbroker.helphome.doc%2Fhelp_home_msgbroker.htm)