<<Interface>>
**wbMain**

| Global Variables | Arguments |
|---|---|
| + TransactionItem: QueueItem<br>+ TransactionData: DataTable<br>+ TransactionNumber: Int32 = 1<br>+ RetryNumber: Int32<br>+ Config: Dictionary<string, object><br>+ SystemError: Exception<br>+ BusinessRuleException: BusinessRuleException<br>+ wbMain_Type: String = in_wbType<br>+ wbMain_Level: Int32 = in_wbLevel<br>+ wbMain_Parent: String = in_wbParent<br>+ wbMain_Audit: Dictionary<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} }<br>+ wbMain_StartTime: DateTime = now() | + in_wbType: String = "MainTask"<br>+ in_wbLevel: Int32 = 0<br>+ in_wbParent: String = string.Empty |

**Legend**

| | |
|---|---|
| blue | input argument |
| red | output argument |
| purple | io argument |

---

**<<State Machine - Initialization State>>**
**wbInitialisation**

Local variables:
+ wbInit_Type: String = in_wbType+", "+"Initialization State"
+ wbInit_Level: Int32 = wbMain_Level+1
+ wbInit_Parent: String = wbMain_Type
+ wbInit_Audit: Dictionary<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} }
+ wbInit_StartTime: DateTime = now()

Try executing Components:

**System - reserved**
+ **InitAllSettings** (in_ConfigFile:"Data\Config.xlsx", in_ConfigSheets:{"Settings", "Constants"}, out_Config:Config)
+ **CloseAllApplications**(in_wbType:in_wbType+", "+"Close All Applications", in_wbParent:wbInit_Type, in_wbLevel:wbInit_Level+1, io_Audit:wbInit_Audit)
+ **KillAllProcesses** ()

**Process**
+ **InitAllApplications** (in_Config:Config, in_wbType: in_wbType+", "+"InitAllApplications", in_wbParent:wbInit_Type, in_wbLevel:wbInit_Level+1, io_Audit:wbInit_Audit)

**Log success**
+ **wbLogging**(in_wbType:wbInit_Type, in_wbLevel:wbInit_Level, in_wbParent:wbInit_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wbInit_Type.toString + " successful", in_wbStart:wbInit_StartTime, in_ChildAudit:wbInit_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

Catch all exceptions, log them, treat or rethrow

**System.Exception: Log failure**
+ **wbLogging**(in_wbType:wbInit_Type, in_wbLevel:wbInit_Level, in_wbParent:wbInit_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbInit_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbInit_StartTime, in_ChildAudit:wbInit_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

---

**<<State Machine - Get Transaction Data State>>**
**wbGetTransactionData**

Local variables:
+ wbGetTransactionData_Type: String = in_wbType+", "+"Get Transaction Data State"
+ wbGetTransactionData_Level: Int32 = wbMain_Level+1
+ wbGetTransactionData_Parent: String = wbMain_Type
+ wbGetTransactionData_Audit: Dictionary<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} }
+ wbGetTransactionData_StartTime: DateTime = now()
+ ShouldStop: Boolean

Try executing Components:

**System - reserved**
+ **OrchestratorStop** ()

**Process**
+ **GetTransactionData** (in_TransactionNumber:TransactionNumber, in_Config:Config, out_TransactionItem:TransactionItem, io_TransactionData:TransactionData)

**Log success**
+ **wbLogging**(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wbGetTransactionData_Type.toString + " successful", in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

Catch all exceptions, log them, treat or rethrow

**System.Exception: Log failure**
+ **wbLogging**(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbGetTransactionData_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

---

**<<State Machine - End Process State>>**
**wbMain**

Local variables:

Try executing Components:

**System - reserved**

**Process**
+ **CloseAllApplications**(in_wbType:in_wbType+", "+"Close All Applications", in_wbParent:wbMain_Type, in_wbLevel:wbMain_Level+1, io_Audit:wbMain_Audit)

**Log success**
+ **wbLogging** (in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, in_wbParent:wbMain_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wbMain_Type.toString + " successful", in_wbStart:wbMain_StartTime, in_ChildAudit:wbMain_Audit, io_ParentAudit:Nothing, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

Catch all exceptions, log them, treat or rethrow

**System.Exception: Log failure**
+ **wbLogging**(in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, in_wbParent:wbMain_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbMain_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbMain_StartTime, in_ChildAudit:wbMain_Audit, io_ParentAudit:Nothing, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

---

**<<State Machine - Process Transaction State>>**
**wbProcess**

Local variables:
+ wbProcessTransaction_Type: String
+ wbProcessTransaction_Level: Int32
+ wbProcessTransaction_Parent: String
+ wbProcessTransaction_Audit: Dictionary<string, object>
+ wbProcessTransaction_StartTime: DateTime

Try executing Components:

**System - reserved**

**Process**
+ **Process** (in_TransactionItem:TransactionItem, in_Config:Config, in_TransactionNumber:TransactionNumber, in_RetryNumber:RetryNumber, in_wbType:in_wbType+", "+"Process Transaction", in_wbLevel:wbProcessTransaction_Level+1, in_wbParent:wbProcessTransaction_Type, io_Audit:wbProcessTransaction_Audit)

**Log success**
+ **wbLogging**(in_wbType:wbProcessTransaction_Type, in_wbLevel:wbProcessTransaction_Level, in_wbParent:wbProcessTransaction_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wbProcessTransaction_Type.toString + " successful", in_wbStart:wbProcessTransaction_StartTime, in_ChildAudit:wbProcessTransaction_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

Catch all exceptions, log them, treat or rethrow

**System.Exception: Log failure**
+ **wbLogging**(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:3, in_LogMessage:"Transaction Number " + TransactionNumber.ToString + ". " + "Work Block "+ wbProcessTransaction_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

**BusinessRuleException: Log failure**
+ **wbLogging**(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:3, in_LogMessage:"Transaction Number " + TransactionNumber.ToString + ". " + "Work Block "+ wbProcessTransaction_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source + ". BRE", in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false))

Finally

**NextTransaction**
+ **NextTransaction**(in_Config:Config, in_SystemError:SystemError, in_BusinessRuleException:BusinessRuleException, in_RetryNumber:RetryNumber, io_TransactionNumber:TransactionNumber, in_TransactionItem:TransactionItem, in_wbType:in_wbType+", "+"Next Transaction", in_wbLevel:wbMain_Level+1, in_wbParent:wbMain_Type, io_Audit:wbMain_Audit)

---

Flow labels: SystemError, BusinessRuleException, Success, Success, SystemError, No Data to Process, New Transaction, Success, BusinessRuleException, SystemError