Flat view (UML) <<Framework Layer>> wbMain Global Variables Arguments + TransactionItem: QueueItem + TransactionData: DataTable + Config: Dictionary<string, object> + in_wbType: String = "MainTask" + SystemError: Exception + in_wbLevel: Int32 = 0 BusinessRuleException: BusinessRuleException + in_wbParent: String = string.Empty + wbMain Type: String = in wbType + in wbFinalExec: Boolean= True + wbMain_Level: Int32 = in_wbLevel + in_wbParentStart: DateTime = now() + wbMain_Parent: String = in_wbParent + in_TaskEnv: String + wbMain_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} } + in_wbPath: String = string.empty + wbMain_StartTime: DateTime = now() + WDMain_Start Time: Date Time = How()
+ SystemReserved: New system.Collections.Generic.Dictionary(Of String, Object) From { {"isQueueItem", False}, {"TransactionNumber", 1}, {"RetryNumber", 0} } + 10_17ansactionData. List<Database |
+ io_Audit: List<Dictionary<string, object>> wMain_FinalExec: Boolean = in_wbFinalExec + wbMain_Path = "Rethrow" + wbMain_HandleError = if(not(string.lsNullOrEmpty(in_wbPath)), in_wbPath+"l"+wbMain_Type, wbMain_Type) SystemError BusinessRuleException Success <<State Machine - Initialization State>> <<State Machine - Get Transaction Data State>> <<Framework Layer>> <<Framework Layer>> wblnitialisation wbGetTransactionData Local variables: Local variables: + wbGetTransactionData Type: String = in wbType+", "+"Get Transaction Data State" + wblnit_Type: String = in_wbType+", "+"Initialization State" + wblnit Level: Int32 = wbMain Level+1 + wbGetTransactionData_Level: Int32 = wbMain_Level+1 + wblnit_Parent: String = wbMain_Type + wbGetTransactionData_Parent: String = wbMain_Type + wblnit_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) from { + wbGetTransactionData_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) {"Successful", 0}, {"Failed", 0} } from { {"Successful", 0}, {"Failed", 0} } + wblnit_StartTime: DateTime = now() + wbGetTransactionData_StartTime: DateTime = now() + wblnit_FinalExec: Boolean = true + wbGetTransactionData_FinalExec: Boolean = true + wblnit_Path = wbMain_Path+"l"+wblnit_Type + ShouldStop: Boolean + wblnit HandleError = "Failed" + wbGetTransactionData_Path = wbMain_Path+"I"+wbGetSetTransactionData_Type + wbGetTransactionData HandleError = "Failed" Try executing Components: Try executing Components: **System - reserved** 包 System - reserved + InitAllSettings (in_ConfigFile:"Data\Config.xlsx", in_ConfigSheets:{"Settings", + OrchestratorStop () "Workblocks", "Tasks", "Constants"}, out_Config:Config) 包 包 **Business Process Layer Interface System - reserved** << Business Process Layer>> << Business Process Layer>> + GetTransactionData + CloseAllApplications (in_wbType:in_wbType+", "+"Close All (in_TransactionNumber:cint(SystemReserved.Item("TransactionNumber,")) in_Config:Conf Applications", in_wbParent:wbInit_Type, in_wbLevel:wbInit_Level+1, out_TransactionItem:TransactionItem, io_TransactionData:TransactionData) io_Audit:wblnit_Audit, in_wbFinalExec: wbMain_FinalExec AND wblnit_FinalExec) 包 Log success **Business Process Layer Interface** << Business Process Layer>> + **wbLogging**(in_wbType:wbGetTransactionData_Type, in wbLevel:wbGetTransactionData Level, in wbParent:wbGetTransactionData Parent, + InitAllApplications (in_Config:Config, in_wbType: in_wbType+", in_LogLevel:1, in_LogMessage:"Work Block "+ wbGetTransactionData_Type.toString + " "+"InitAllApplications", in_wbParent:wblnit_Type, in_wbLevel:wblnit_Level+1, successful", in_wbStart:wbGetTransactionData_StartTime, io_Audit:wblnit_Audit, in_wbFinalExec: wbMain_FinalExec AND wblnit_FinalExec) in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND 起 BusinessRuleException is Nothing, true, false), in_wbFinalExec: wbMain_FinalExec AND Log success wbGetTransactionData FinalExec) + wbLogging(in_wbType:wblnit_Type, in_wbLevel:wblnit_Level, in_wbParent:wbInit_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ Catch all exceptions, log them, treat or rethrow wblnit_Type.toString + " successful", in_wbStart:wblnit_StartTime, 包 System.Exception: Log failure in ChildAudit:wbInit Audit, io ParentAudit:wbMain Audit, in SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, + **wbLogging**(in_wbType:wbGetTransactionData_Type, true, false), in wbFinalExec: wbMain FinalExec AND wbInit FinalExec) 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, Catch all exceptions, log them, treat or rethrow in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, 起 System. Exception: Log failure in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false), in_wbFinalExec: wbMain_FinalExec AND + wbLogging(in_wbType:wblnit_Type, in_wbLevel:wblnit_Level, wbGetTransactionData FinalExec) in_wbParent:wbInit_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wblnit_Type.toString + " failed with exception "+exception.Message+" at Source **New Transaction** "+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, <<State Machine - Process Transaction State>> true, false), in_wbFinalExec: wbMain_FinalExec AND wbInit_FinalExec) <<Framework Layer>> Success wbProcess Local variables: + wbProcessTransaction_Type: String + wbProcessTransaction_Level: Int32 + wbProcessTransaction_Parent: String + wbProcessTransaction_Audit: Dictionarry<string, object> + wbProcessTransaction_StartTime: DateTime + wbProcessTransaction_FinalExec: Boolean = true <<State Machine - End Process State>> + wbProcessTransaction_Path = wbMain_Path+"I"+wbProcessTransaction_Type << Framework Layer>> + wbProcessTransaction_HandleError = "Failed" wbMain Try executing Components: Local variables: 起 **System - reserved** Try executing Components: 包 **Business Process Layer Interface** 起 **System - reserved** << Business Process Layer>> + **Process** (in TransactionItem:TransactionItem, in Config:Config, **Business Process Layer Interface** in_TransactionNumber:SystemReserved.Item("TransactionNumber").ToString, in_RetryNumber:SystemReserved.Item("RetryNumber").ToString, << Business Process Layer>> in_wbType:in_wbType+", "+"Process Transaction", in_wbLevel:wbProcessTransaction_Level+1, in_wbParent:wbProcessTransaction_Type, + CloseAllApplications(in_wbType:in_wbType+", "+"Close All Applications", in_wbParent:wbMain_Type, in_wbLevel:wbMain_Level+1, io_Audit:wbProcessTransaction_Audit, in_wbFinalExec: wbMain_FinalExec AND wbProcessTransaction FinalExec) o_Audit:wbMain_Audit, in_wbFinalExec: wbMain_FinalExec) 包 Log success Log success + wbLogging(in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, + wbLogging(in_wbType:wbProcessTransaction_Type, in_wbParent:wbMain_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ in_wbLevel:wbProcessTransaction_Level, in_wbParent:wbProcessTransaction_Parent, wbMain_Type.toString + " successful", in_wbStart:wbMain_StartTime, in_LogLevel:1, in_LogMessage:"Work Block "+ wbProcessTransaction_Type.toString + " _ChildAudit:wbMain_Audit, io_ParentAudit:Nothing, in_SuppressLoggi successful", in_wbStart:wbProcessTransaction_StartTime, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, in_ChildAudit:wbProcessTransaction_Audit, io_ParentAudit:wbMain_Audit, true, false), in_wbFinalExec: wbMain_FinalExec) in SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false), in_wbFinalExec: wbMain_FinalExec AND wbProcessTransaction_FinalExec) Catch all exceptions, log them, treat or rethrow System. Exception: Log failure Catch all exceptions, log them, treat or rethrow + wbLogging(in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, 起 System. Exception: Log failure in_wbParent:wbMain_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbMain_Type.toString + " failed with exception "+exception.Message+" at Source + wbLogging(in_wbType:wbGetTransactionData_Type, "+exception.Source, in wbStart:wbMain StartTime, in ChildAudit:wbMain Audit, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, io_ParentAudit:Nothing, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError in_LogLevel:3, in_LogMessage:"Transaction Number " + TransactionNumber.ToString + " is Nothing AND BusinessRuleException is Nothing, true, false), in_wbFinalExec: " + "Work Block "+ wbProcessTransaction_Type.toString + " failed with exception wbMain FinalExec) "+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), in_wbFinalExec: wbMain_FinalExec AND Legend wbProcessTransaction_FinalExec) input argument output argument io argument 包 **NextTransaction** + **NextTransaction**(in_Config:Config, in_SystemError:SystemError, in BusinessRuleException:BusinessRuleException, io 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, in_wbFinalExec: wbMain_FinalExec AND wbProcessTransaction FinalExec)