

4. PMX Classes

You can find the list of methods and fields of the Produmex WMS classes in this section.

4.1. General

4.1.1. SboRecordsetHelper

Fields	-	-
Methods	ISboRecordset RunQuery(ILog log, string query, PmxDbConnection dbConn)	Provide the result of the query into ISboRecordset class

4.1.2. ISboRecordset

Fields	EoF	Bool
Methods	void MoveFirst()	-
	void MoveLast()	-
	void MoveNext()	-
	void MovePrevious()	-
	void RedoOriginalQuery()	-
	GetTypedValue<string>("<Col_Name>") GetTypedValue<int>("Col_Name") GetTypedValue<double>("Col_Name")	Read data from the recordset

4.1.3. PmxItemAllConnectionsProvider

Initialization	new PmxItemAllConnectionsProvider (conn);
Methods	PmxItemInfo GetCachedItemInfo (string itemCode)
	PmxItemInfo GetItemInfo (string itemCode)

4.1.4. PmxItemInfo

Fields	<pre> private string ItemCode; private bool IsLogisticCarrier; private bool IsLogisticUnit; private bool IsReturnableItem; private string Description; private bool HasBestBeforeDate; private bool IsInventoryItem; private bool HasBatchnumber; private string CodeBars; private int? ItemLabelReportKey; private int NumberOfCopiesItemLabel; private bool NeedsReasonForPurchaseReturn; private bool NeedsReasonForSalesReturn; private string QualityStatusCodeProduction; private string QuarantinedQualityStatusCodeReception; private string ReleasedQualityStatusCodeReception; private string QualityStatusCodeReceptionController; private string QualityStatusCodeSalesReturn; private bool ScanBaseComponent; private PmxItemInfo m_baseItemInfo; private int ShelfLifeInDays; private string ExpiryDefinitionCodeForProduction; private string ExpiryDefinitionCodeForReception; private string InventoryUom; private string m_purchaseUom; private string SalesUom; private double NumberOfItemsPerPurchaseUnit; private double NumberOfItemsPerSalesUnit; private string SalesBarcode; private string PurchaseBarcode; private bool PrintItemLabel; private string Uom2; private double DefaultQuantityUom2; private bool CorrectStockUom; private bool CorrectStockUom2; private bool UseUom2; private PmxUomToUse UomToUseForPurchase; private PmxUomToUse UomToUseForInventoryTransitions; private PmxUomToUse UomToUseForSales; private int UomDecimals; private int Uom2Decimals; private bool HasSecondBatchNumber; private string PictureName; private string PackingImage; private string PackingRemarks; private string VendorItemDescription; private string CustomItemDescription; private bool HasNoValue; private string LowestSellablePackagingType; private int ShelfLifeInDaysReception; private string SerialNumberFormat; private bool CreateSsccOnReception; private Collection<PmxItemPackagingTypeInfo> ListOfPackagingTypes = new Collection<PmxItemPackagingTypeInfo>(); private PmxDictionary<string, double> BarcodesAndUomQuantities = new PmxDictionary<string, double>(); private PmxDictionary<string, double> PurchaseBarcodesAndUomQuantities = new PmxDictionary<string, double>(); private PmxDictionary<string, double> SalesBarcodesAndUomQuantities = new PmxDictionary<string, double>(); private PmxDictionary<string, string> DefaultWarehouseLocationOrZone = new PmxDictionary<string, string>(); </pre>
---------------	---

Methods	-
----------------	---

4.1.5. PmxItemTransactionalInfoProvider

Initialization	new PmxItemTransactionalInfoProvider (conn);	-
Methods	void ChangeBBDOnBatch (int itriKey, DateTime newBBD, bool changeExpDate = false)	Change BBD of a batch number itriKey = PMX_ITRI.InternalKey
	void ChangeInternalBatchNumber (int itriKey, string newBatchNumber2)	Change 2nd Batch number of a batch number itriKey = PMX_ITRI.InternalKey

4.1.6. PmxLogisticUnitIDProvider

Initialization	new PmxLogisticUnitIDProvider(conn);	-
Methods	int GenerateNewLogisticUnit (string supplierPalletNumber)	Generates a new SSCC, the result of the method is PMX_LUID.InternalKey
	bool CheckIsLuidsInInventory (Collection<int> luids)	-
	bool CheckIsSSCCMasterLogisticUnit (string scc)	-
	int GetLUIDBySSCC (string scc)	PMX_LUID.InternalKey

4.1.7. PmxOseBin

Fields	<pre>private string Code public string Name public bool IsActive public bool IsPickLocation public int Sequence public bool CanBeLinedUp public double? MaximumQuantity public int? MaximumLogisticUnits public bool AllowCountDuringCycleCount public bool AllowCountDuringOtherOperation public bool NeedsToBeCounted public int? LockedBy public int CountAfterNumberOfDays public int CountAfterNumberOfOperations</pre>
Methods	-

4.1.8. PmxOseBinProvider

Initialization	new PmxOseBinProvider (conn);
Methods	PmxOseBin GetB0(string key)

4.1.9. LogisticUnitGoodsReceipt

```
public class LogisticUnitGoodsReceipt
```

```
{
AllowReceptionWithoutLUID =
logisticUnitGoodsReceipt.AllowReceptionWithoutLUID,
IsMoveToLockedStockLocation =
logisticUnitGoodsReceipt.IsMoveToLockedStockLocation,
ItemsOnLogisticUnit =
Mapper.LocalCollectionOfLogisticUnitItemGoodReceipts(logisticUnitGoodsReceipt.ItemsOnLogisticUnit),
LogisticUnitId = logisticUnitGoodsReceipt.LogisticUnitId,
MasterLUID = logisticUnitGoodsReceipt.MasterLUID,
Reason = logisticUnitGoodsReceipt.Reason,
SpecificLocationCode = logisticUnitGoodsReceipt.SpecificLocationCode,
SSCC = logisticUnitGoodsReceipt.SSCC,
SupplierPalletNumber = logisticUnitGoodsReceipt.SupplierPalletNumber,
UnitPrice = logisticUnitGoodsReceipt.UnitPrice,
}
```

4.1.10. LogisticUnitItemGoodReceipt

```
public class LogisticUnitItemGoodReceipt
{
AutoSelectP0 = logisticUnitItemGoodReceipt.AutoSelectP0,
BatchNumber1 = logisticUnitItemGoodReceipt.BatchNumber1,
BatchNumber2 = logisticUnitItemGoodReceipt.BatchNumber2,
BeasItemVersion = logisticUnitItemGoodReceipt.BeasItemVersion,
BestBeforeDate = logisticUnitItemGoodReceipt.BestBeforeDate,
FullListOfPackagingTypes =
Mapper.LocalCollectionOfPackagingTypeInfoos(logisticUnitItemGoodReceipt.FullListOfPackagingTypes),
IsLogisticCarrier = logisticUnitItemGoodReceipt.IsLogisticCarrier,
IsLogisticCarrierForLogisticUnit =
logisticUnitItemGoodReceipt.IsLogisticCarrierForLogisticUnit,
ItemCode = logisticUnitItemGoodReceipt.ItemCode,
ListOfBatchAttributes =
Mapper.LocalCollectionOfItemTransactionalBatchAttributeInfos(logisticUnitItemGoodReceipt.ListOfBatchAttributes),
ListOfPackagingTypes =
Mapper.LocalCollectionOfItemTransactionalPackagingTypeInfoos(logisticUnitItemGoodReceipt.ListOfPackagingTypes),
ListOfReasonsByZoneType =
Mapper.LocalDictionaryOfReasonInfos(logisticUnitItemGoodReceipt.ListOfReasonsByZoneType),
ListOfSerialNumbers = logisticUnitItemGoodReceipt.ListOfSerialNumbers,
OverrideLocationCode = logisticUnitItemGoodReceipt.OverrideLocationCode,
PurchaseDocRef =
Mapper.LocalDocumentRef(logisticUnitItemGoodReceipt.PurchaseDocRef),
PurchaseDocumentLineNum =
logisticUnitItemGoodReceipt.PurchaseDocumentLineNum,
QuantitiesForUom2 = logisticUnitItemGoodReceipt.QuantitiesForUom2,
Quantity = logisticUnitItemGoodReceipt.Quantity,
}
```

```
QuantityPerUom = logisticUnitItemGoodReceipt.QuantityPerUom,  
QuantityUom2 = logisticUnitItemGoodReceipt.QuantityUom2,  
ReasonInfo = logisticUnitItemGoodReceipt.ReasonInfo,  
UnitPrice = logisticUnitItemGoodReceipt.UnitPrice,  
Uom2 = logisticUnitItemGoodReceipt.Uom2,  
Usage = logisticUnitItemGoodReceipt.Usage,  
}
```

4.1.11. PackagingTypeInfo

```
public class PackagingTypeInfo  
{  
private string m_packagingTypeCode;  
private string m_packagingTypeName; // Uom  
private double m_quantity;  
private double? m_initialQuantity;  
private double m_quantityPerPack = 1;  
private Collection<string> m_barcode;  
private bool m_showOnScreen;  
private int m_numberOfDecimals = 0;  
private bool m_hideDuringEnteringQuantity;  
}
```

4.2. Move

Customization Articles for WMS scripting:
[How to change the Quality Status by scripting](#)
[How to remove the SSCC for certain items, and put them on the location directly](#)

4.2.1. PmxMoveProvider types

Initialization	NEW PmxMoveProvider(conn);	-
Methods	PmxMoveProvider GetNewBO()	Generate a new Business Object into memory
	string AddBO (PmxMove bo)	Create the object in database
	PmxMoveLine GetNewAddedLine (PmxMove document)	Add a new line for the item

4.2.2. PmxMove types

No need to do any configuration on this object.

Example usage:

```
PmxMoveProvider moveProv = new PmxMoveProvider(conn);
PmxMove move
= moveProv.GetNewB0();
PmxMoveLine moveLine = moveProv.GetNewAddedLine(
move
);
moveLine.ItemCode = ...;
moveLine.Quantity = ...;
...;
moveProv.AddB0(
move
, true);
```

4.2.3. PmxMoveLine types**Fields**

```
private string SourceStorageLocationCode;
private int? SourceLogisticUnitIdentificationKey;
private string DestinationStorageLocationCode;
private int? DestinationLogisticUnitIdentificationKey;
private string SourceQualityStatusCode;
private string DestinationQualityStatusCode;
private bool isLogisticCarrier;
private int? ItemTransactionalInfoKey;
private string ReasonCode;
private string ReasonFreeText;
private string ReasonLocationCode;
```

4.2.4. Enumeration types

```
public enum PmxMoveOrderType
{
    Move = 1,
    PutAway = 2,
    Replenish = 4,
    WarehouseTransfer = 8,
    PutAwayProduction = 12
}
```

```
public enum PmxMoveOrderStatus
{
    NothingMoved = 1,
    Closed = 2,
    PartiallyMoved = 4
}
```

```
public enum PmxMoveInOneTime
{
    Invalid = 0,
    CannotBeMovedInOneTime = 1,
    CanBeInOneTime = 2,
    MustBeMovedInOneTime = 4
}
```

```
public enum PmxMoveOrderStockLevel
{
    Detail = 1,
    Item = 2,
    MasterLuid = 4
}
```

4.2.5. PmxMoveOrderProvider

Initialization	<code>new PmxMoveOrderProvider(conn);</code>	-
Methods	<code>PmxMoveOrder GetNewB0()</code>	Generate a new Business Object into memory
	<code>string AddB0 (PmxMoveOrder bo)</code>	Create the object in database
	<code>void UpdateB0 (PmxMoveOrder bo, bool onlyUpdateHeader, bool baseDocumentIsClosing)</code>	Updates the Business object in database
	<code>void CloseDocument (PmxMoveOrder document)</code>	Closing Move Order document
	<code>PmxMoveOrderLine GetNewAddedLine (PmxMoveOrder document)</code>	Add a new line to the document

4.2.6. PmxMoveOrder

Fields	<pre>private PmxMoveOrderStatus MoveOrderStatus; private DateTime DueDate; private int Priority; private PmxMoveOrderType MoveOrderType; private PmxMoveInOneTime MoveLogUnitIn1Time; private int? LockedBy; private string FromPmxWhsCode; private string ToPmxWhsCode; private string Remarks;</pre>
Methods	-

4.2.7. PmxMoveOrderLine

Fields	<pre>private PmxMoveOrderStatus MoveOrderLineStatus; private string SourceStorageLocationCode; private int? SourceLogisticUnitIdentificationKey; private string DestinationStorageLocationCode; private int? DestinationLogisticUnitIdentificationKey; private string QualityStatusCode; private int? ItemTransactionalInfoKey; private PmxMoveOrderStockLevel StockLevel; private string WaBoxCode;</pre>
Methods	-

4.3. Move Order

Customization Articles for WMS scripting:
[How to generate Move Order by scripting](#)

4.4. Picklist Proposal

4.4.1. Enumeration types

```
public enum PmxPickListProposalStockStatus
{
    None,
    Partially,
    All
}
```

```
public enum PmxPickObjectType
{
    Sales,
    WhsTransfer,
    Production,
    WhsTransferProd
}
```

```
public enum PmxInventoryLockingLevel
{
    ItemNoLocking = 0,
    ItemQuality = 1,
    ItemBatchNumberBestBeforeDate = 2,
    ItemLUID = 4,
    ItemDetail = 8
}
```

4.4.2. PmxPickListProposalProvider

Initialization	<code>new PmxPickListProposalProvider (conn);</code>	-
Methods	<code>PmxPickListProposal GetNewBO()</code>	Generate a new Business Object into memory
	<code>PmxPickListProposal GetBO(int key)</code>	-
	<code>string AddBO (PmxPickListProposal bo)</code>	Create the object in database
	<code>void UpdateBO (PmxPickListProposal bo, bool onlyUpdateHeader, bool baseDocumentIsClosing)</code>	Updates the Business object in database
	<code>void CloseDocument (PmxPickListProposal document)</code>	Closing Move Order document
	<code>PmxMoveOrderLine GetNewAddedLine (PmxPickListProposal document)</code>	Add a new line to the document

4.4.3. PmxPickListProposal

Fields	<pre>private string CardCode; private string CardName; private string ShipToCode; private string ShipToaddress; private string DestinationStorageLocation; private PmxPickListProposalStockStatus FullStockStatus; private PmxPickListProposalStockStatus NotExpiredStockStatus; private DateTime DueDate; private string PickPackRemarks; private string Remarks; private int? RouteLineDocEntry; private int? RouteLineLineNum; private bool IsCustomerCollect; private string PickListType; private BusinessObjectProperty<int?> ShippingID; private string MoveToWarehouse; private string MoveToLocationCode; private PmxPickObjectType PickObjType;</pre>
Methods	-

4.4.4. PmxPickListProposalLine

Fields	<pre>private int? ItemTransactionalInfoKey; private int? LogisticUnitIdentificationKey; private string QualityStatusCode; private PmxPickListProposalStockStatus FullStockStatus; private PmxPickListProposalStockStatus NotExpiredStockStatus; private double PickListQuantity; private PmxInventoryLockingLevel InvLockLevel; private bool ForceBatch; private bool IsSampleOrder; public double? QuantityForTempLock;</pre>
Methods	-

4.5. Picklist

4.5.1. Enumeration types

```
public enum PmxPickListStatus
{
    NotReady = 1,
    Closed = 2,
    PartiallyReady = 4,
    Ready = 8,
    PartiallyPicked = 0x10,
    Picked = 0x20,
    PartiallyDelivered = 0x40,
    PartiallyPacked = 0x80,
    Packed = 0x100,
    ForcedClosed = 0x200,
    PartiallyShipped = 0x400,
    Shipped = 0x800
}
```

```
public enum PmxPickObjectType
{
    Sales,
    WhsTransfer,
    Production,
    WhsTransferProd
}
```

```
public enum PmxInventoryLockingLevel
{
    ItemNoLocking = 0,
    ItemQuality = 1,
    ItemBatchNumberBestBeforeDate = 2,
    ItemLUID = 4,
    ItemDetail = 8
}
```

4.5.2. PmxPickListProvider

Initialization	<code>new PmxPickListProposalProvider (conn);</code>	-
-----------------------	--	---

Methods	<code>PmxPickListProposal GetNewB0()</code>	Generate a new Business Object into memory
	<code>PmxPickListProposal GetB0(int key)</code>	-
	<code>string AddB0 (PmxPickListProposal bo)</code>	Create the object in database
	<code>void UpdateB0 (PmxPickListProposal bo, bool onlyUpdateHeader, bool baseDocumentIsClosing)</code>	Updates the Business object in database
	<code>void CloseDocument (PmxPickListProposal document)</code>	Closing Move Order document
	<code>PmxMoveOrderLine GetNewAddedLine (PmxPickListProposal document)</code>	Add a new line to the document

4.5.3. PmxPickList

Fields	<pre>private PmxPickListStatus PickListStatus; private string CardCode; private string ShipToAddress; private string ShipToCode; private string DestStorLocCode; private int? PickListProposalEntry; private BusinessObjectProperty<int> Priority; private DateTime DueDate; private int? LockBy; private string CardName; private int? RouteKey; private string ReasonCodeNotFullShipping; private string ReasonFreeTextNotFullShipping; private int? WaveKey; private string PickPackRemarks; private bool IsCustomerCollect; private bool IsPrinted; private string PickListType; private DateTime? LastPrintDateTime; private int? PackLockBy; private string MoveToWarehouse; private string MoveToLocationCode; private PmxPickObjectType PickObjType;</pre>
Methods	-

4.5.4. PmxPickListLine

Fields	<pre>private PmxPickListStatus PickListLineStatus; private int? ItemTransactionalInfoKey; private string StorageLocationCode; private int? LogisticUnitIdentificationKey; private string QualityStatusCode; private int Sequence; private double QuantityPicked; private double QuantityPacked; private double OriginalQuantity; private double? QuantityPickedUom2; private double? QuantityPackedUom2; private string ReasonCodeNotFullPicking; private PmxInventoryLockingLevel InvLockLevel; private bool ForceBatch; private bool IsSampleOrder;</pre>
Methods	-

4.6. Print

Customization Articles for WMS scripting:
[How to trigger a print event from a script](#)

4.6.1. PmxReportProvider

Initialization	<code>new PmxReportProvider (conn);</code>	-
Methods	<code>PmxReport GetB0(int key)</code>	The code of the report from the OSE

4.6.2. PmxOsePrinterProvider

Initialization	<code>new PmxOsePrinterProvider (conn);</code>	-
Methods	<code>PmxOsePrinter GetNewB0()</code>	Not supported
	<code>PmxOsePrinter GetB0(string key)</code>	The code of the printer from the OSE

The method provides the closest printer to a location.
 Page size of the printer must be configured in the call. DeviceID can be null.

```
PmxOsePrinter GetPrinterForLocation( string CurrentLocationCode, string DeviceID, string PageSizeCode)
```

4.6.3. ReportPrinterDevice

Initialization	<code>new ReportPrinterDevice (conn);</code>	-
-----------------------	--	---

Methods	<pre>void PrintReport (PmxReport report, PmxOsePrinter printer, CultureInfo cultureInfo, int numberOfCopies, DataSet ds, Collection<ReportParameter> reportParameters) --- CR parameters from PMX layouts</pre>	Create a new object for the printing
----------------	--	--------------------------------------

4.6.4. Example

```
PmxReportProvider reportProvider = new PmxReportProvider(conn);
PmxReport report = reportProvider.GetBO(6); // report code FROM OSE

PmxOsePrinterProvider printerProvider = new PmxOsePrinterProvider(conn);
PmxOsePrinter printer = printerProvider.GetNewBO();
printer = printerProvider.GetBO("PRINTER"); // printer code from OSE

// create the report parameter structure
Collection<ReportParameter> reportParameters = new
Collection<ReportParameter>();
reportParameters.Add(new ReportParameter("@luid", Luids[i]));

ReportPrinterDevice device = new ReportPrinterDevice(conn);
device.PrintReport(report, printer, null, 1, null, reportParameters);
```

4.7 Cycle Count

4.7.1 PmxCycleCountProvider

- Initialization
 - new PmxCycleCountProvider(conn);
- Methods
 - PerformCountDuringCycleCount(Collection<CycleCountInventoryItemInfo> listOfItemsToUpdate, string locationCode, string deviceId)
 - Performs the count during cycle count.

```
<param name="listOfItemsToUpdate">The list of items to
update.</param> <param name="locationCode">The location
code.</param>
```

4.7.2 CycleCountInventoryItemInfo

- Initialization
 - new CycleCountInventoryItemInfo();

- Class to hold the data for cycle count inventory items

4.7.3 Example

```
ReasonInfo reasonInfo = new ReasonInfo();
Collection<CycleCountInventoryItemInfo> listOfItemsToCount = new
Collection<CycleCountInventoryItemInfo>();
PmxCycleCountProvider prov = new PmxCycleCountProvider(conn);
CycleCountInventoryItemInfo invItemInfo = new CycleCountInventoryItemInfo();
PmxItemAllConnectionsProvider provItem = new
PmxItemAllConnectionsProvider(conn);

PmxItemInfo itemInfo = provItem.GetCachedItemInfo("I001");
invItemInfo.ItemInfo = itemInfo;
invItemInfo.ItemCode = "I001";
invItemInfo.ItemDescription = "I001 - I001";
invItemInfo.CurrentStock = 10;
invItemInfo.Difference = -2; // change the stock to 8
invItemInfo.DifferenceUom2 = null;
invItemInfo.QualityStatusCode = "RELEASED";
invItemInfo.ReasonInfo = reasonInfo;
invItemInfo.BatchNumber = null;
invItemInfo.SecondBatchNumber = null;
invItemInfo.BestBeforeDate = null;
invItemInfo.SSCC = null;
invItemInfo.LocationCode = "003";
invItemInfo.Uom = null;
invItemInfo.Uom2 = null;

listOfItemsToCount.Add(invItemInfo);
prov.PerformCountDuringCycleCount(listOfItemsToCount, "003", null);
```

From:

<https://wiki.produmex.name/> - **Produmex**

Permanent link:

https://wiki.produmex.name/doku.php?id=implementation:wms:wms_scripting_site:pmx_classes

Last update: **2026/03/04 10:29**

