
FinProduction.DLL

Purpose: this application is an ActiveX DLL, which will be used to process production receipts, back-flush , consume forecast., repetitive production, repetitive scrap and start production .

Installation:

- 1) Install the Pervasive client.
- 2) Run InstalFinproduction.EXE file
- 3) Make the following MAX DLL's available:

MAX DLL'S
MAXEXEC2 DLL
MAXBTRV2 DLL
MAXTRAN2 DLL

These files will be installed by the MAX Client Setup routine in \MAX\MFW2 and will reside in the server in \EXACT\MAX\NET\UPDATES\MFW2. As MAX is updated it is important to also update the files that are used by theFinProduction.DLL.

Usage:

- 1) From MS Access or Visual Basic add a reference to FinProduction.DLL
- 2) Add a class declaration: Private FPro As ProcessMAXTrans
- 3) Set a class instance: Set FPro = New ProcessMAXTrans

This application has the following six functions:

StartUp(ShowMessages as boolean) as Boolean

This function is called at the beginning to open the MAX database and to initiate MAXUpdate. If MAX has not been installed on this workstation a pop-up window will appear the first time this function is called which will ask for the location of the MAX data and executables.

Sample Syntax:

```
If FPro.Startup(ShowMessages) then  
    ....  
endif
```

The parameter 'Showmessages' = True/False depending on whether you want to see error messages displayed as they occur.



FinProduction.DLL

ProdReceipt(PartID as string, SerialNumber as string ,Masterpack as string, BinID as string) as Boolean

This function will do the following:

1. A Lot ID representing a Master Pack, will be assigned for every 25 units of production. The Lot ID will have the following format: MPXX0000001. XX will represent the Bin ID. The last 7 digits of the Lot will be incremented sequentially as they are created. The Part being produced must be under both Lot and Serial control. When a new Lot is generated the ID will be returned in the MasterPack parameter.
2. The product will be received into the parts' default stock location.
3. The components for the part will be back-flushed from their default stock locations.
4. A matching MS order will be looked for with a date within the current week and will be reduced as production is reported.

Sample Syntax:

```
MasterPack = ""
If FPro.ProdReceipt(PartID), SerialNumber, MasterPack) Then
  If MasterPack > "" Then
    MPack.Caption = MasterPack
    MPack.Refresh
  End If
End If
```

RepReceipt(UserID, PartID As String, Workcenter As String, qty As Double, LotID, SerID) As Boolean

This function will do the following:

- 1) Receive the Part into the default stock location.
- 2) Issue (un-planned) all components required by the part produced from their respective default locations.

Sample Syntax:

```
MasterPack = ""
If FPro.RepReceipt("Tester", PartID, "EASSY", 1, "", "") Then
End If
```



FinProduction.DLL

RepScrap(UserID, PartID As String, WorkcenterAS As String, OpSeq, qty As Double) As Boolean

This function will do the following:

1) Issue all components indicated in the Part Routing, that were picked up to the operation sequence as indicated in the part routing.

Sample Syntax:

```
If FPro.Repscrap("Tester", Text1(0), Text1(2), Text1(3), 1) Then  
End
```

StartProd(UserID, Workcenter, PartID) As Boolean

This function will do the following:

Write a zero quantity Receipt transaction to indicate the start of a unit.

Sample Syntax:

```
If FPro.startprod("Tester", WorkCenter, PartID) Then  
End If
```

ShuDown() as Boolean

This function will close the MAX database and stop MAXUpdate.

Sample Syntax:

```
If FPro.ShutDown() Then  
End If
```

