**Copy ALL CODE below into a txt document and save the file extension as AutoGenerateSupplierID.bpm**

<?xml version="1.0" encoding="utf-8"?><directiveGroup xsi:noNamespaceSchemaLocation="Msg.Epicor.Mfg.Bpm.DirectiveGroup.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><name>Auto Supplier ID</name><version>9.05.607</version><boSourceGroup boSource="BO"><methods><method><code>Vendor.GetNewVendor</code><businessObject>Vendor</businessObject><name>GetNewVendor</name><preProcDirectives /><baseProcDirectives /><postProcDirectives><directive><name>Autopopulate Supplier ID</name><order>10</order><isEnabled>True</isEnabled><companyIndependent>False</companyIndependent><preventDeadloops>False</preventDeadloops><reenterMax>1000</reenterMax><conditionText /><actionText>synchronously execute ABL DEFINE VAR ivendornum AS INTEGER. ... record nothing</actionText><globalID>b6aa695f-254c-eabb-e111-e320ad86fa37</globalID><isUpToDate>True</isUpToDate><scope /><conditions><sourceCode>/\*Add source code that should be executed before the designed conditions here\*/
{&amp;CALL\_DESIGNED\_CONDITIONS}
/\*Add source code that should be executed after the designed conditions here\*/</sourceCode></conditions><actions><sourceCode>/\*Add source code that should be executed before the designed actions here\*/
{&amp;CALL\_DESIGNED\_ACTIONS}
/\*Add source code that should be executed after the designed actions here\*/</sourceCode><action><order>1</order><typeName>FourGlActionType</typeName><terminateOnErr>False</terminateOnErr><data><part><order>1</order><text>&lt;Root&gt;&lt;SystemVersion&gt;2.5.607.0&lt;/SystemVersion&gt;&lt;CustomVersion&gt;FourGlActionCfg3&lt;/CustomVersion&gt;&lt;Body&gt;&amp;lt;SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&amp;gt;

&amp;lt;SOAP-ENV:Body&amp;gt;

&amp;lt;a1:FourGlActionConfig2 id="ref-1" xmlns:a1="http://schemas.microsoft.com/clr/nsassem/Epicor.Mfg.UI.Bpm.Configurations/Epicor.Mfg.Bpm.BPMPlugins"&amp;gt;

&amp;lt;ProgramActionConfig2\_x0060\_1\_x002B\_core href="#ref-4"/&amp;gt;

&amp;lt;/a1:FourGlActionConfig2&amp;gt;

&amp;lt;a2:FourGlActionCfg3 id="ref-4" xmlns:a2="http://schemas.microsoft.com/clr/nsassem/Bpm.CodeGen.Config/Epicor.Mfg.Bpm.CodeGen"&amp;gt;

&amp;lt;Declarations id="ref-5"&amp;gt;&amp;lt;/Declarations&amp;gt;

&amp;lt;IsForward&amp;gt;false&amp;lt;/IsForward&amp;gt;

&amp;lt;Procedure id="ref</text></part><part><order>2</order><text>-6"&amp;gt;DEFINE VAR ivendornum AS INTEGER.
DEFINE VAR cvendorid AS CHARACTER.
FOR EACH ttvendor:
FIND FIRST company WHERE company.Company = CUR-COMP NO-LOCK.
ASSIGN ivendornum = company.number03 + 1.
ASSIGN cvendorid = STRING(ivendornum).
ASSIGN ttvendor.vendorid = cvendorid.
RUN lib\UpdateTableBuffer.p(INPUT BUFFER company:HANDLE,&amp;amp;#39;number03&amp;amp;#39;,ivendornum).
END.&amp;lt;/Procedure&amp;gt;

&amp;lt;SourceProc href="#ref-7"/&amp;gt;

&amp;lt;AsyncMode&amp;gt;BPM\_ASYNC\_NONE&amp;lt;/AsyncMode&amp;gt;

&amp;lt;DoRecordCall&amp;gt;false&amp;lt;/DoRecordCall&amp;gt;

&amp;lt;DoRecordReturn&amp;gt;false&amp;lt;/DoRecordReturn&amp;gt;

&amp;lt;batchOption&amp;gt;OnceForAllMatching&amp;lt;/batchOption&amp;gt;

&amp;lt;primaryTable href="#ref-5"/&amp;gt;

&amp;lt;Version xsi:null="1"/&amp;gt;

&amp;lt;/a2:FourGlActionCfg3&amp;gt;

&amp;lt;a2:MethodInfoCfg2 id="ref-7" xmlns:a2="http://schemas.microsoft.com/clr/nsassem/Bpm.CodeGen.Config/Epicor.Mfg.Bpm.CodeGen"&amp;gt;

&amp;lt;BoSource id="ref-8"&amp;gt;BO&amp;lt;/BoSource&amp;gt;

&amp;lt;ObjName id="ref-9"&amp;gt;Vendor&amp;lt;/ObjName&amp;gt;

&amp;lt;Parameters href=</text></part><part><order>3</order><text>"#ref-10"/&amp;gt;

&amp;lt;Location id="ref-11"&amp;gt;\GetNewVendor.p&amp;lt;/Location&amp;gt;

&amp;lt;Version id="ref-12"&amp;gt;Bpm.CodeGen.Config.MethodInfoCfg2&amp;lt;/Version&amp;gt;

&amp;lt;/a2:MethodInfoCfg2&amp;gt;

&amp;lt;SOAP-ENC:Array id="ref-10" SOAP-ENC:arrayType="a2:ParamCfg[1]" xmlns:a2="http://schemas.microsoft.com/clr/nsassem/Bpm.CodeGen.Config/Epicor.Mfg.Bpm.CodeGen"&amp;gt;

&amp;lt;item href="#ref-13"/&amp;gt;

&amp;lt;/SOAP-ENC:Array&amp;gt;

&amp;lt;a2:ParamCfg id="ref-13" xmlns:a2="http://schemas.microsoft.com/clr/nsassem/Bpm.CodeGen.Config/Epicor.Mfg.Bpm.CodeGen"&amp;gt;

&amp;lt;Direction id="ref-14"&amp;gt;INPUT-OUTPUT&amp;lt;/Direction&amp;gt;

&amp;lt;Name id="ref-15"&amp;gt;VendorDataSet&amp;lt;/Name&amp;gt;

&amp;lt;Type id="ref-16"&amp;gt;DATASET$16&amp;lt;/Type&amp;gt;

&amp;lt;Version xsi:null="1"/&amp;gt;

&amp;lt;/a2:ParamCfg&amp;gt;

&amp;lt;/SOAP-ENV:Body&amp;gt;

&amp;lt;/SOAP-ENV:Envelope&amp;gt;

&lt;/Body&gt;&lt;/Root&gt;</text></part></data><sourceCode><part><block>0</block><order>1</order><text>run ttVendor\_Backup.

run ttVendorAttch\_Backup.

run ttEntityGLC\_Backup.

run ttTaxExempt\_Backup.

run ttVendRestriction\_Backup.

run ttVendBank\_Backup.

run ttVendBankAttch\_Backup.

run ttVendCntMain\_Backup.

run ttVenMFBill\_Backup.

run ttVendorPP\_Backup.

run ttVenPPMFBill\_Backup.

run ttVenPPUPSEmail\_Backup.

run ttVendPPRestriction\_Backup.

run ttVendCnt\_Backup.

run ttVendCntAttch\_Backup.

run ttVenUPSEmail\_Backup.

run GetNewVendor{RESULT} (INPUT-OUTPUT dataset VendorDataSetM).

run ttVendor\_Restore.

run ttVendor\_copyFromMatch.

run ttVendorAttch\_Restore.

run ttVendorAttch\_copyFromMatch.

run ttEntityGLC\_Restore.

run ttEntityGLC\_copyFromMatch.

run ttTaxExempt\_Restore.

run ttTaxExempt\_copyFromMatch.

run ttVendRestriction\_Restore.

run ttVendRestriction\_copyFromMatch.

run ttVendBank\_Restore.

run ttVendBank\_copyFromMatch.

run ttVendBankAttch\_Restore.

run ttVendBankAttch\_copyFromMatch.

run ttVendCntMain\_Restore.

run ttVendCntMain\_copyFromMatch.

run ttVenMFBill\_</text></part><part><block>0</block><order>2</order><text>Restore.

run ttVenMFBill\_copyFromMatch.

run ttVendorPP\_Restore.

run ttVendorPP\_copyFromMatch.

run ttVenPPMFBill\_Restore.

run ttVenPPMFBill\_copyFromMatch.

run ttVenPPUPSEmail\_Restore.

run ttVenPPUPSEmail\_copyFromMatch.

run ttVendPPRestriction\_Restore.

run ttVendPPRestriction\_copyFromMatch.

run ttVendCnt\_Restore.

run ttVendCnt\_copyFromMatch.

run ttVendCntAttch\_Restore.

run ttVendCntAttch\_copyFromMatch.

run ttVenUPSEmail\_Restore.

run ttVenUPSEmail\_copyFromMatch.</text></part><part><block>1</block><order>1</order><text>

/\* AblActionGen {

\*/

procedure GetNewVendor{RESULT}:

define input-output parameter dataset for VendorDataSet.

{&amp;TRY\_PRIVATE}

/\* ProcedureGen {

\*/

DEFINE VAR ivendornum AS INTEGER.
DEFINE VAR cvendorid AS CHARACTER.
FOR EACH ttvendor:
FIND FIRST company WHERE company.Company = CUR-COMP NO-LOCK.
ASSIGN ivendornum = company.number03 + 1.
ASSIGN cvendorid = STRING(ivendornum).
ASSIGN ttvendor.vendorid = cvendorid.
RUN lib\UpdateTableBuffer.p(INPUT BUFFER company:HANDLE,'number03',ivendornum).
END.

/\*

} ProcedureGen \*/

{&amp;CATCH\_PRIVATE}

end procedure.

/\*

} AblActionGen \*/</text></part></sourceCode></action></actions></directive></postProcDirectives></method></methods></boSourceGroup><externalData /></directiveGroup>