**Steps:**

1. In table ‘Inventory Setup’ id: 313, add the following fields.
2. HS Code Format, Code 20, id: 80025
3. HS Code Prefix Length, Integer, id: 80030

Then pull to its page id: 461

1. In table ‘Tariff Number’ id: 260, add a new field.
2. Show HS Code, Code 20, 80025

Table ‘Tariff Number’ in “No.”

|  |
| --- |
| **No. - OnValidate()**  **// Sambidh HS Code July 8 2024**  **InvSetup.GET;**  **format := InvSetup."HS Code Format";**  **check := CompareHSCode("No.", format);**  **IF NOT check THEN BEGIN**  **ERROR('The entered HS Code for item I-00003 is not in agreement with the format specified in your HS Code Format.');**  **END;**  **"No." := UPPERCASE("No.");**  **"Show HS Code" := COPYSTR**("No.", 1, InvSetup."HS Code Prefix Length");  // Sambidh HS Code July 8 2024  **CompareHSCode(HSCode : Code[20];Format : Text[20]) : Boolean**  **// Sambidh HS Code July 8 2024**  **Check := TRUE;**  **IF STRLEN(HSCode) = STRLEN(Format) THEN**  **FOR i:=1 TO STRLEN(HSCode) DO BEGIN**  **Cf := COPYSTR(Format, i, 1);**  **Ce := COPYSTR(HSCode, i, 1);**  **CASE Cf OF**  **'#':**  **IF NOT ((Ce >= '0') AND (Ce <= '9')) THEN**  **Check := FALSE;**  **'@':**  **IF STRPOS(Text005, UPPERCASE**(Ce)) = 0 THEN  Check := FALSE;  ELSE  IF NOT ((Cf = Ce) OR (Cf='?')) THEN  Check := FALSE  END;  END  ELSE  Check := FALSE;  EXIT(Check);  // Sambidh HS Code July 8 2024 |

Reset Action in Page 310 Tariff Number

|  |
| --- |
| **Reset - OnAction()**  **InvSetup.RESET;**  **InvSetup.GET;**  **Prefix := InvSetup."HS Code Prefix Length";**  **TariffNumber.RESET;**  **IF TariffNumber.FINDSET THEN BEGIN**  **REPEAT**  **"Show HS Code" := COPYSTR**("No.", 1, Prefix);  TariffNumber.MODIFY;  UNTIL TariffNumber.NEXT = 0;  END |

|  |
| --- |
| // Code by Sambidh for HS Code  //"HS Code" := Item."Tariff No.";  IF TariffNumber.GET(Item."Tariff No.") THEN  "HS Code" := TariffNumber."Show HS Code"; |

Sales Line in “No.”

**For SUBISU(BC):**

TariffNumber table extension

|  |
| --- |
| tableextension 50082 TariffNumberExt extends "Tariff Number"  {      fields      {          field(4; "Show HS Code"; Code[20])          {              Caption = '';              DataClassification = ToBeClassified;          }          modify("No.")          {              trigger OnAfterValidate()              var                  myInt: Integer;                  check: Boolean;                  format: Text[20];              begin                  InvSetup.Get();                  format := InvSetup."HS Code Format";                  check := CompareHSCode("No.", format)              end;          }      }      var          InvSetup: Record "Inventory Setup";          text0005: Label 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';      local procedure CompareHSCode(HSCode: Text[20]; Format: Text[20]): Boolean      var          myInt: Integer;          i: Integer;          Cf: Text[1];          Ce: Text[1];          Check: Boolean;      begin          Check := true;          if StrLen(HSCode) = StrLen(Format) then              for i := 1 to StrLen(HSCode) do begin                  Cf := CopyStr(Format, i, 1);                  Ce := CopyStr(HSCode, i, 1);                  case Cf of                      '#':                          if not ((Ce >= '0') and (Ce <= '9')) then                              Check := false;                      '@':                          if StrPos(text0005, UpperCase(Ce)) = 0 then                              Check := false;                      else                          if not ((Cf = Ce) or (Cf = '?')) then                              Check := false                  end;              end          else              Check := false;          exit(Check);      end;  } |

Sales Line table extension

|  |
| --- |
| tableextension 50040 "Sales Line" extends "Sales Line"  {      fields      {          field(50000; "HS Code"; Code[20])          {              DataClassification = ToBeClassified;          }          modify("No.")          {              trigger OnAfterValidate()              var                  myInt: Integer;                  TariffNumber: Record "Tariff Number";                  Item: Record Item;              begin                  if TariffNumber.Get(Item."Tariff No.") then                      "HS Code" := TariffNumber."Show HS Code";              end;          }      }  } |

Page extension of TariffNumber

|  |
| --- |
| pageextension 50142 "Tariff NumbersExt" extends "Tariff Numbers"  {      actions      {          addlast(Processing)          {              action(Reset)              {                  Caption = 'Reset';                  ApplicationArea = All;                  trigger onAction()                  var                      myInt: Integer;                      InvSetup: Record "Inventory Setup";                      Prefix: Integer;                      TariffNumber: Record "Tariff Number";                  begin                      InvSetup.Reset;                      InvSetup.Get;                      Prefix := InvSetup."HS Code Prefix Length";                      TariffNumber.Reset;                      if TariffNumber.Findset then begin                          repeat                              Rec."Show HS Code" := Copystr(Rec."No.", 1, Prefix);                              TariffNumber.Modify;                          until TariffNumber.next = 0;                      end;                  end;              }          }      }  } |