

Spécificité Technique

Gestion des palettes

- ⇒ Modifications global des labels. Edition du fichier XML 'langage.config'.
- ⇒ Ajouter un champ 'RefSage' dans la table TReferenceProduitBase

	Id	RefExpertive	RefFoum	Libelle	Poids	AControler
1	8	39SL0020S	9SL0020S	Blanc Transparent	20	0
2	9	39SP1000S	9SP1000S	Jaune de Base	20	0

- ⇒ Ajouter une propriété 'RefSage'

```
public class Reference
{
    public int Id { get; set; }
    public string RefExpertive { get; set; }
    public string RefSupplier { get; set; }
    public string Label { get; set; }
    public double Weight { get; set; }
    public bool Controlled { get; set; }
    public string refExpAndLabel
    {
        get { return RefExpertive + " - " + Label; }
    }
}
```

- ⇒ Modifier les noms des propriétés pour RefSage1 jusqu'à RefSage8.

```
public class DTOPaletteReference
{
    public int Id { get; set; }
    public string Number { get; set; }
    public string RefExpertive1 { get; set; }
    public string RefExpertive2 { get; set; }
    public string RefExpertive3 { get; set; }
    public string RefExpertive4 { get; set; }
    public string RefExpertive5 { get; set; }
    public string RefExpertive6 { get; set; }
    public string RefExpertive7 { get; set; }
    public string RefExpertive8 { get; set; }
```

- ⇒ Ajouter une assignation à la propriété 'RefSage'.

```

private static Reference loadReference (DataRow row)
{
    Reference reference = new Reference();

    if (row["Id"].ToString() != DBNull.Value.ToString())
        reference.Id = (int)row["Id"];
    if (row["RefExpertive"].ToString() != DBNull.Value.ToString())
        reference.RefExpertive = (string)row["RefExpertive"];
    if (row["RefFourn"].ToString() != DBNull.Value.ToString())
        reference.RefSupplier = (string)row["RefFourn"];
    if (row["Libelle"].ToString() != DBNull.Value.ToString())
        reference.Label = (string)row["Libelle"];
}

```

⇒ Modifier les assignments des propriétés.

```

if (table.Rows[j]["ref1"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive1 = (string)table.Rows[j]["ref1"];
if (table.Rows[j]["ref2"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive2 = (string)table.Rows[j]["ref2"];
if (table.Rows[j]["ref3"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive3 = (string)table.Rows[j]["ref3"];
if (table.Rows[j]["ref4"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive4 = (string)table.Rows[j]["ref4"];
if (table.Rows[j]["ref5"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive5 = (string)table.Rows[j]["ref5"];
if (table.Rows[j]["ref6"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive6 = (string)table.Rows[j]["ref6"];
if (table.Rows[j]["ref7"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive7 = (string)table.Rows[j]["ref7"];
if (table.Rows[j]["ref8"].ToString() != DBNull.Value.ToString())
    palette.RefExpertive8 = (string)table.Rows[j]["ref8"];

```

⇒ Modifier les requêtes SQL afin de retourner les éléments du champ « RefSage » depuis la table TReferenceProduitBase.

```

GO
/***** Object: StoredProcedure [dbo].[SEL_ALL_PALETTEREFERENCE]    Script Date: 12/09/2016 08:51:09 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[SEL_ALL_PALETTEREFERENCE]
    @NumeroPalette varchar(50),
    @CdeFournisseur varchar(10),
    @dateDeb datetime,
    @dateEnd datetime,
    @printed int,
    @refId int,
    @refFab varchar(20),
    @AffQte int,
    @LocId int
AS
    declare @sql nvarchar(max)

    SET @sql = 'SELECT [dbo].[TPalette].',
    SET @sql = @sql + ' reference1.RefExpertive as ref1,reference1.Reffourn as Reffourn1,reference2.RefExpertive as ref2,reference2.Reffourn as Re',
    SET @sql = @sql + ' FROM [dbo].[TPalette]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference1 ON reference1.Id=[IdReferenceFab1]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference2 ON reference2.Id=[IdReferenceFab2]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference3 ON reference3.Id=[IdReferenceFab3]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference4 ON reference4.Id=[IdReferenceFab4]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference5 ON reference5.Id=[IdReferenceFab5]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference6 ON reference6.Id=[IdReferenceFab6]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference7 ON reference7.Id=[IdReferenceFab7]'
    SET @sql = @sql + ' LEFT JOIN dbo.TReferenceProduitBase as reference8 ON reference8.Id=[IdReferenceFab8]'
    SET @sql = @sql + ' INNER JOIN dbo.TLocalisation ON dbo.TLocalisation.Id=[dbo].[TPalette].TLocalisationId'
    SET @sql = @sql + ' WHERE 1=1'
    IF @NumeroPalette<>' '
        BEGIN
            SET @sql = @sql + ' AND dbo.TPalette.Numero=''' + @NumeroPalette + ''''
        END
    IF @CdeFournisseur<>' '
        BEGIN
            SET @sql = @sql + ' AND dbo.TPalette.CommandeFrs=''' + @CdeFournisseur + ''''
        END
    END

```

Gestion d'une palette

- ⇒ Modifier l'affichage du contenu des combobox « Référence »
(Concaténer : Ref Fournisseur + libellé).

```
int index = this.cbref1.SelectedIndex;
this.cbref1.DataSource = new BindingList<Reference>(lstRef);
this.cbref1.DisplayMember = "refExpAndLabel";
this.cbref1.ValueMember = "refExpAndLabel";
if (index <= this.cbref1.Items.Count - 1)
    this.cbref1.SelectedIndex = index;
index = this.cbref2.SelectedIndex;
this.cbref2.DataSource = new BindingList<Reference>(lstRef);
this.cbref2.DisplayMember = "refExpAndLabel";
this.cbref2.ValueMember = "refExpAndLabel";
this.cbref2.SelectedIndex = index;
```

```
public string refExpAndLabel
{
    get { return RefExpertive + " - " + Label; }
}
```

- ⇒ Modifier le label associé à la sélection du combobox (Afficher le code Sage)
Changer la valeur des labels afin d'afficher la référence Sage.

```
private void cbref1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(this.cbref1.Text))
        if (((Reference)this.cbref1.SelectedItem) != null)
            this.lblRef1.Text = ((Reference)this.cbref1.SelectedItem).RefSupplier;
        else
            this.lblRef1.Text = "⚡";
    else
        this.lblRef1.Text = "⚡";
}
```

FmanageReference

- ⇒ Modification Interface du terme 'Expertive'.
- ⇒ Ajouter un contrôle TextBox «txtRefSage »
- ⇒ Ajouter l'assignation au contrôle TextBox « txtRefSage»

```
private void loadreference ()
{
    this.txtlabel.Text = this.CurrentReference.Label;
    this.txtPoids.Text = Convert.ToDouble(this.CurrentReference.Weight).ToString();
    this.txtRefExpertive.Text = this.CurrentReference.RefExpertive;
    this.txtRefFrs.Text = this.CurrentReference.RefSupplier;
    this.FMANAGEREFERENCE_chkYes.Checked = this.CurrentReference.Controlled;
    this.FMANAGEREFERENCE_chkNo.Checked = !this.CurrentReference.Controlled;
}
```

- ⇒ Ajouter l'assignation à la propriété 'RefSage' depuis le contrôle « txtRefSage »

```
private void cmdSave_Click(object sender, EventArgs e)
{
    if (this.CheckValidation())
    {
        this.CurrentReference.Label = this.txtlabel.Text;
        this.CurrentReference.RefExpertive = this.txtRefExpertive.Text;
        this.CurrentReference.RefSupplier = this.txtRefFrs.Text;
        this.CurrentReference.Weight = Convert.ToDouble(this.txtPoids.Text);
        this.CurrentReference.Controlled = this.FMANAGEREFERENCE_chkYes.Checked;
        _servReference.save(this.CurrentReference);
        this.Close();
    }
}
```

- ⇒ Modifier les requêtes SQL afin d'insérer / update une référence Sage.

```
ALTER PROCEDURE [dbo].[INS_REFERENCE]
    @RefExpertive varchar(20),
    @RefFourn varchar(50),
    @Libelle varchar(100),
    @poids float,
    @AControler bit,
    @Id int output
AS
INSERT INTO [dbo].[TReferenceProduitBase]
    ([RefExpertive]
    ,[RefFourn]
    ,[Libelle]
    ,[Poids]
    ,AControler)
VALUES
    (@RefExpertive
    ,@RefFourn
    ,@Libelle
    ,@poids
    ,@AControler)
SET @Id = @@IDENTITY
```

```

USE [DEVIntraEncreTD]
GO
/***** Object:  StoredProcedure [dbo].[UPD_REFERENCE]    Script Date: 09/09/2016 16:35:58 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[UPD_REFERENCE]
    @RefExpertive varchar(20),
    @RefFourn varchar(50),
    @Libelle varchar(100),
    @poids float,
    @AControler bit,
    @Id int
AS

UPDATE [dbo].[TReferenceProduitBase]
SET [RefExpertive] = @RefExpertive
, [RefFourn] = @RefFourn
, [Libelle] = @Libelle
, [Poids] = @poids
, AControler = @AControler
WHERE Id=@Id

```

⇒ Ajouter un paramètre 'RefSage'

```

private static Reference update(Reference reference)
{
    IQuery requete =
        new Query(System.Configuration.ConfigurationManager.AppSettings["dbBaseIntrasystem"].ToString(),
            "[dbo].[UPD_REFERENCE]");
    requete.AddParameter("@RefExpertive", reference.RefExpertive, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@RefFourn", reference.RefSupplier, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@Libelle", reference.Label, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@poids", reference.Weight, DbType.Double, ParameterDirection.Input);
    requete.AddParameter("@AControler", reference.Controlled, DbType.Boolean, ParameterDirection.Input);
    requete.AddParameter("@Id", reference.Id, DbType.Double, ParameterDirection.Input);
    requete.Execute();
    return reference;
}

private static Reference insert(Reference reference)
{
    IQuery requete =
        new Query(System.Configuration.ConfigurationManager.AppSettings["dbBaseIntrasystem"].ToString(),
            "[dbo].[INS_REFERENCE]");
    requete.AddParameter("@RefExpertive", reference.RefExpertive, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@RefFourn", reference.RefSupplier, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@Libelle", reference.Label, DbType.String, ParameterDirection.Input);
    requete.AddParameter("@poids", reference.Weight, DbType.Double, ParameterDirection.Input);
    requete.AddParameter("@AControler", reference.Controlled, DbType.Boolean, ParameterDirection.Input);
    requete.AddParameter("@Id", 0, DbType.Double, ParameterDirection.Output);
    requete.Execute();
    reference = get(Convert.ToInt32(requete.GetAttributOut("@Id")));
    return reference;
}

```

Liste des entrées / sorties

- ⇒ Modification Interface du terme 'Expertive' pour 'Sage'
- ⇒ Ajouter la propriété 'RefSage' dans la class DTOMovement.
- ⇒ Ne pas affiché la propriété « Numreference » dans le contrôle (DTGV) DataGridView.

```

public class DTOMovement
{
    public int Id { get; set; }
    public string NumPalette { get; set; }
    public string Numreference { get; set; }
    public string NumreferenceSupplier { get; set; }

```

⇒ Ajouter une assignation à la propriété 'RefSage'

```
for (int j = 0; j < table.Rows.Count; j++)
{
    DTOMovement mouvement = new DTOMovement();

    if (table.Rows[j]["Id"].ToString() != DBNull.Value.ToString())
        mouvement.Id = Convert.ToInt32(table.Rows[j]["Id"]);
    if (table.Rows[j]["Numero"].ToString() != DBNull.Value.ToString())
        mouvement.NumPalette = (string)table.Rows[j]["Numero"];
    if (table.Rows[j]["RefExpertise"].ToString() != DBNull.Value.ToString())
        mouvement.Numreference = (string)table.Rows[j]["RefExpertise"];
    if (table.Rows[j]["RefFourn"].ToString() != DBNull.Value.ToString())
        mouvement.NumreferenceSupplier = (string)table.Rows[j]["RefFourn"];
    if (table.Rows[j]["DateScan"].ToString() != DBNull.Value.ToString())
        mouvement.DateScan = (DateTime)table.Rows[j]["DateScan"];
}
```

⇒ Modifier les requêtes SQL afin de retourner les éléments du champ « RefSage » de la table TReferenceProduitBase

```
ALTER PROCEDURE [dbo].[SEL_ALL_MOUVEMENT]
    @NumeroPalette varchar(50),
    @NumeroRef varchar(20),
    @Entry int,
    @OF varchar(6),
    @SaisieExpertise int,
    @dateDeb datetime,
    @dateEnd datetime
AS
declare @sql nvarchar(max)

SET @sql = 'SELECT [dbo].[TScanPalette].[Id],dbo.TPalette.Numero,dbo.TReferenceProduitBase.RefExpertise,[DateScan]'
SET @sql = @sql + ',[MouvementEntre],[OF],[dbo].[TScanPalette].[Poids],[nbSeau],InExpertise,RefFourn'
SET @sql = @sql + ' FROM [dbo].[TScanPalette]'
SET @sql = @sql + ' INNER JOIN dbo.TReferenceProduitBase ON dbo.TReferenceProduitBase.Id=[dbo].[TScanPalette].[IdReference]'
SET @sql = @sql + ' INNER JOIN dbo.TPalette ON dbo.TPalette.Id=[dbo].[TScanPalette].[IdPalette]'
SET @sql = @sql + ' WHERE 1=1'
```

Entrée / Sortie

⇒ Supprimer le terme 'Expertise' de l'interface.

⇒ Modifier le nom du contrôle « txtRefExpertise » pour « txtRef ».

⇒ Modifier les conditions pour une égalité possible avec une référence Sage.

```
if (!string.IsNullOrEmpty(palette.Reference1.RefExpertise))
    if (palette.Reference1.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference1;
if (!string.IsNullOrEmpty(palette.Reference2.RefExpertise))
    if (palette.Reference2.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference2;
if (!string.IsNullOrEmpty(palette.Reference3.RefExpertise))
    if (palette.Reference3.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference3;
if (!string.IsNullOrEmpty(palette.Reference4.RefExpertise))
    if (palette.Reference4.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference4;
if (!string.IsNullOrEmpty(palette.Reference5.RefExpertise))
    if (palette.Reference5.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference5;
if (!string.IsNullOrEmpty(palette.Reference6.RefExpertise))
    if (palette.Reference6.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference6;
if (!string.IsNullOrEmpty(palette.Reference7.RefExpertise))
    if (palette.Reference7.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference7;
if (!string.IsNullOrEmpty(palette.Reference8.RefExpertise))
    if (palette.Reference8.RefExpertise.ToUpper() == this.txtRefExpertise.Text.ToUpper())
        reference = palette.Reference8;
```

