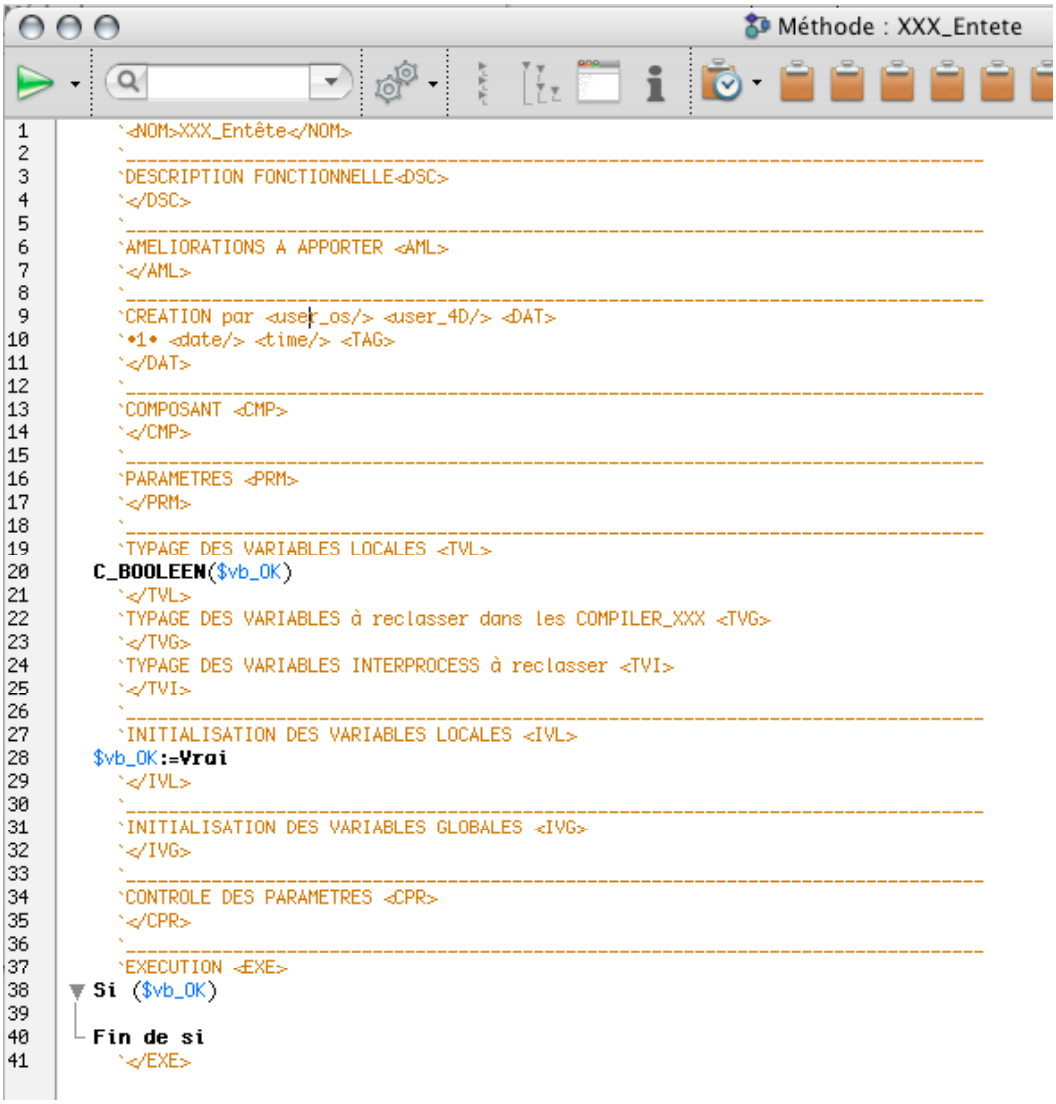
A. PRINCIPES

La macro est minimale et ne contient qu'une ligne de code ; elle appelle une méthode en lui passant 1 ou 2 paramètres : le nom de la méthode qui contient tags et commentaires, la 2e en option contient les commentaires et le code à insérer entre les tags.

La date, l'heure et l'utilisateur sont remplacés.

B. EXEMPLE DE MACRO D'EN-TETE



La macro contient des tags qui servent notamment à la déclaration des variables.

C. MACRO

```
//============================================================================
//===== SEPARATEUR ===========================================================
//============================================================================
<macro name="-zz----------">
        <text> </text>
</macro>
```

```
//=============================================================================
//===== METHODE FORMULAIRE DIALOGUE ===========================================
//=============================================================================
<macro name="MF Methode Dialogue">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFG_Dialogue")</method>
</text>
</macro>


//=============================================================================
//===== METHODE FORMULAIRE LISTE ==============================================
//=============================================================================
<macro name="MF Formulaire Liste">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFG_Liste")</method>
</text>
</macro>


//=============================================================================
//===== METHODE FORMULAIRE PAGE ===============================================
//=============================================================================
<macro name="MF Formulaire Page">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFG_Page")</method>
</text>
</macro>


//=============================================================================
//===== MFD METHODE SPECIFIQUE DIALOGUE =======================================
//=============================================================================
<macro name="MFD Methode specifique Dialogue">
<text>
`<NOM><Method_name/></NOM>
```

```
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFD_Tttt")</method>
</text>
</macro>


//=============================================================================
//===== MFD METHODE SPECIFIQUE LISTE ==========================================
//=============================================================================
<macro name="MFD Methode specifique Liste">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFL_Tttt")</method>
</text>
</macro>


//=============================================================================
//===== MFD METHODE SPECIFIQUE PAGE ===========================================
//=============================================================================
<macro name="MFD Methode specifique Page">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MFP_Tttt")</method>
</text>
</macro>


//=============================================================================
//===== METHODE MENU APPEL TABLE ==============================================
//=============================================================================
<macro name="MMT Methode Menu Appel Table">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MMT_Tttt")</method>
</text>
</macro>


//=============================================================================
//===== METHODE SPECIFIQUE TABLE ==============================================
//=============================================================================
```

```
<macro name="MST Methode Specifique Table">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MST_Tttt")</method>
</text>
</macro>


//=============================================================================
//===== METHODE INITIALISATION VARIABLES DU MODULE ============================
//=============================================================================
<macro name="MIV Module">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_Entete";"XXX_MIV")</method>
</text>
</macro>


//=============================================================================
//===== METHODE COMPILATION VARIABLES DU MODULE ===============================
//=============================================================================
<macro name="COMPILER_Module_XXX">
<text>
`<NOM><Method_name/></NOM>
<method>MCR_Methode_type_Coller("XXX_COMPILER_Module_XXX")</method>
</text>
</macro>
```

D. METHODES

1. MCR_Méthode_type_Coller

```
`<NOM>MCR_Méthode_type_Coller</NOM>
`_____________________________________________________________________________
`DESCRIPTION FONCTIONNELLE <DSC>
`Cette méthode appelle la méthode type et la copie dans la méthode ouverte
`</DSC>
`_____________________________________________________________________________
`AMELIORATIONS A APPORTER <AML>
`Coller la sélection
`</AML>
`_____________________________________________________________________________
`CREATION par Bernard Escaich <DAT>
`•1• 28/04/07 12:19:58 <TAG>
`•2• 30/04/07 08:50:04
`•3• 28/09/07 00:09:59
`</DAT>
`_____________________________________________________________________________
```

```
   `COMPOSANT <CMP>
   `</CMP>
   `_____________________________________________________________________________________
   `PARAMETRES <PRM>
   `Variables entrée
C_TEXTE($1)   `$1 : Nom de la méthode contenant l'en-tête type
C_TEXTE($2)   `$2 : Nom de la méthode contenant le code-type à insérer entre les balises EXE
C_TEXTE(_textSel)   `<--TEXTE SELECTIONNE(vide si>32000 cars.)
C_BLOB(_blobSel)   `<--TEXTE SELECTIONNE
C_ENTIER LONG(_selLen)   `<--Longueur du texte sélectionné
C_TEXTE(_textMethod)   `<--Texte de la méthode(vide si>32000 cars.)
C_BLOB(_blobMethod)   `<--Texte de la méthode
C_ENTIER LONG(_methodLen)   `<--Longueur du texte de la méthode
   `Variables sortie
C_TEXTE(_textReplace)   `-->Résultat à substituer,""par défaut
C_BLOB(_blobReplace)   `-->Résultat à substituer,""par défaut
C_ENTIER LONG(_action)   `Type de substitution dans la méthode d'appel : 0=Aucun, 1=Insérer _textReplace
   `2=Insérer _blobReplace, 3=Remplacer la méthode par _textReplace, 4=Remplacer la méthode par _blobReplace
   `</PRM>
   `_____________________________________________________________________________________
   `TYPAGE DES VARIABLES LOCALES <TVL>
C_BOOLEEN($vb_OK)
C_TEXTE($vt_Méthode_en_tête)
C_TEXTE($vt_Méthode_type)
TABLEAU TEXTE($tt_Méthode_lignes;0)
   `</TVL>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES LOCALES <IVL>
$vb_OK:=Faux
$vt_Méthode_en_tête:=$1
Si (Nombre de parametres>=2)
      $vt_Méthode_type:=$2
Fin de si
   `</IVL>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES GLOBALES <IVG>
TABLEAU TEXTE($tt_Méthode_lignes;0)
_action:=4   `Remplacement de la méthode par le blob _blobReplace
FIXER TAILLE BLOB(_blobReplace;0)
   `</IVG>
   `_____________________________________________________________________________________
   `CONTROLE DES PARAMETRES <CPR>
   `</CPR>
   `_____________________________________________________________________________________
   `EXECUTION <EXE>
Si (Vrai)
      Si (Nombre de parametres>=2)
         MCR__Methode_inserer (->_blobReplace;$1;$2)
      Sinon
         MCR__Methode_inserer (->_blobReplace;$1)
      Fin de si
Fin de si
   `</EXE>
```

2. MCR Méthode insérer

```
   `<NOM>MCR__Méthode_insérer</NOM>
   `_____________________________________________________________________________________
   `DESCRIPTION FONCTIONNELLE <DSC>
   `</DSC>
   `_____________________________________________________________________________________
   `AMELIORATIONS A APPORTER <AML>
   `Récupérer la description
   `</AML>
   `_____________________________________________________________________________________
   `CREATION par Bernard Escaich<DAT>
   `•1• 03/05/07 22:23:01
   `</DAT>
```

```
   `
   `_____________________________________________________________________________________
   `COMPOSANT <CMP>
   `</CMP>
   `
   `_____________________________________________________________________________________
   `PARAMETRES<PRM>
C_POINTEUR($1)    `$1 : Blob à remplir
C_TEXTE($2)    `$2 : Nom de la méthode contenant l'en-tête type
Si (Nombre de parametres>=3)
      C_TEXTE($3)    `$3 : Nom de la méthode contenant le code-type à insérer entre les balises EXE
Fin de si
   `</PRM>
   `
   `_____________________________________________________________________________________
   `TYPAGE DES VARIABLES LOCALES <TVL>
C_ALPHA(4;$va4_Res_Type)
C_BLOB($vx_CC4D_rsrc_Blob_token)
C_BLOB($vx_CC4D_rsrc_Blob_token2)
C_BOOLEEN($vb_Insérer)
C_BOOLEEN($vb_OK)    `INITIALISATION DES VARIABLES LOCALES <IVL>
C_ENTIER LONG($i)
C_ENTIER LONG($j)
C_ENTIER LONG($k)
C_ENTIER LONG($l)
C_ENTIER LONG($vl_Blob_taille)
C_ENTIER LONG($vl_CC4D_ID)
C_ENTIER LONG($vl_CC4D_ID2)
C_ENTIER LONG($vl_errCode)
C_ENTIER LONG($vl_Longueur)
C_TEXTE($vt_Balise_début)
C_TEXTE($vt_Balise_fin)
C_TEXTE($vt_Méthode_en_tête)
C_TEXTE($vt_Méthode_ligne1)
C_TEXTE($vt_Méthode_ligne2)
C_TEXTE($vt_Méthode_type)
TABLEAU TEXTE($tt_Méthode_ligne;0)
TABLEAU TEXTE($tt_Méthode_ligne1;0)
TABLEAU TEXTE($tt_Méthode_ligne2;0)
   `</TVL>
   `TYPAGE DES VARIABLES à reclasser dans les COMPILER_XXX <TVG>  `</TVG>
   `TYPAGE DES VARIABLES INTERPROCESS à reclasser <TVI>
   `C_ALPHA(2;◊CR)`INITIALISATION DES VARIABLES LOCALES <IVL>
   `</TVI>
   `
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES LOCALES <IVL>
$vb_OK:=Vrai
$vt_Méthode_en_tête:=$2
Si (Nombre de parametres>=3)
      $vt_Méthode_type:=$3
Fin de si
$va4_Res_Type:="CC4D"
$vb_Insérer:=Vrai
   `</IVL>
   `
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES GLOBALES <IVG>
   `</IVG>
   `
   `_____________________________________________________________________________________
   `CONTROLE DES PARAMETRES <CPR>
   `</CPR>
   `
   `_____________________________________________________________________________________
   `EXECUTION <EXE>
Si ($vb_OK)

      `Recherche de la ressource contenant le texte de la méthode d'en-tête
   $vl_CC4D_ID:=MTD_MethodList_Meth_CC4D_id_get ($vt_Méthode_en_tête)
      `Chargement de ma méthode tokenisée dans un blob
   $vl_errCode:=API_Resource_Get ($va4_Res_Type;$vl_CC4D_ID;->$vx_CC4D_rsrc_Blob_token)
      `Detokenizing the blob (method tokenized) into a text array containing the lines of the method
   MTD_MethodDetokenize ($vx_CC4D_rsrc_Blob_token;->$tt_Méthode_ligne1)
```

```
Si (Nombre de parametres>=3)
        `Recherche de la ressource contenant le texte de la méthode-type
    $vl_CC4D_ID2:=MTD_MethodList_Meth_CC4D_id_get ($vt_Méthode_type)
        `Chargement de ma méthode tokenisée dans un blob
    $vl_errCode:=API_Resource_Get ($va4_Res_Type;$vl_CC4D_ID2;->$vx_CC4D_rsrc_Blob_token2)
        `Detokenizing the blob (method tokenized) into a text array containing the lines of the method
    MTD_MethodDetokenize ($vx_CC4D_rsrc_Blob_token2;->$tt_Méthode_ligne2)
Fin de si


Si ($vl_errCode=0)
    Boucle ($i;1;Taille tableau($tt_Méthode_ligne1))
        $vt_Méthode_ligne1:=$tt_Méthode_ligne1{$i}+◊CR
            `
            `Balises à remplacer par des valeurs
        Si ($vt_Méthode_ligne1="@<user_os/>@")
            $vt_Méthode_ligne1:=Remplacer chaine($vt_Méthode_ligne1;"<user_os/>";"")    `Pas de commande
4D ???
        Fin de si
        Si ($vt_Méthode_ligne1="@<user_4D/>@")
            $vt_Méthode_ligne1:=Remplacer chaine($vt_Méthode_ligne1;"<user_4D/>";Utilisateur courant)
        Fin de si
        Si ($vt_Méthode_ligne1="@<date/>@")
            $vt_Méthode_ligne1:=Remplacer chaine($vt_Méthode_ligne1;"<date/>";Chaine(Date du jour;Spécial
forcé ))
        Fin de si
        Si ($vt_Méthode_ligne1="@<time/>@")
            $vt_Méthode_ligne1:=Remplacer chaine($vt_Méthode_ligne1;"<time/>";Chaine(Heure courante))
        Fin de si
        Si ($vt_Méthode_ligne1="@<caret/>@")
            $vt_Méthode_ligne1:=Remplacer chaine($vt_Méthode_ligne1;"";"")
        Fin de si

            `Balises à remplacer par le contenu de la 2e méthode
            `Détecter la balise de début et celle de fin
        Si ($vt_Méthode_type#"")
                `On colle, mais on inhibe le collage au tour suivant jusqu'à l'apparition de la balise de fin
            Au cas ou
                : ($vt_Méthode_ligne1="@<DSC>@")
                    $vt_Balise_début:="<DSC>"
                    $vt_Balise_fin:="</DSC>"

                : ($vt_Méthode_ligne1="@<AML>@")
                    $vt_Balise_début:="<AML>"
                    $vt_Balise_fin:="</AML>"

                : ($vt_Méthode_ligne1="@<EXE>@")
                    $vt_Balise_début:="<EXE>"
                    $vt_Balise_fin:="</EXE>"

            Fin de cas

            Si ($vt_Balise_début#"")
                $vt_Méthode_ligne1:=$tt_Méthode_ligne1{$i}+◊CR
                $vl_Longueur:=Longueur($vt_Méthode_ligne1)
                $vl_Blob_taille:=Taille BLOB($1->)
                FIXER TAILLE BLOB($1->;$vl_Blob_taille+$vl_Longueur)
                Boucle ($j;0;$vl_Longueur-1)
                    $1->{$vl_Blob_taille+$j}:=Code ascii($vt_Méthode_ligne1≤$j+1≥)
                Fin de boucle
            Fin de si

               `On zappe la méthode 1 jusqu'à la balise de fin correspondante
            Si ($vt_Balise_début#"")
                Repeter
```

```
                    $i:=$i+1
                    $vt_Méthode_ligne1:=$tt_Méthode_ligne1{$i}+◊CR
                Jusque ($vt_Méthode_ligne1=("@"+$vt_Balise_fin+"@"))
            Fin de si

        Fin de si

        `On insère les lignes de la méthode 2 compris entre les balises
        Si ($vt_Balise_début#"")
            $j:=Chercher dans tableau($tt_Méthode_ligne2;"@"+$vt_Balise_début+"@")
            Si ($j>=1)
                `Chercher la balise de fin
                $k:=Chercher dans tableau($tt_Méthode_ligne2;"@"+$vt_Balise_fin+"@";$j)
                Si ($k>$j)
                    `On insère toutes les lignes comprises entre $j et $k
                    Boucle ($l;$j+1;$k-1)
                        $vt_Méthode_ligne2:=$tt_Méthode_ligne2{$l}+◊CR
                        $vl_Longueur:=Longueur($vt_Méthode_ligne2)
                        $vl_Blob_taille:=Taille BLOB($1->)
                        FIXER TAILLE BLOB($1->;$vl_Blob_taille+$vl_Longueur)
                        Boucle ($j;0;$vl_Longueur-1)
                            $1->{$vl_Blob_taille+$j}:=Code ascii($vt_Méthode_ligne2≤$j+1≥)
                        Fin de boucle
                    Fin de boucle
                Fin de si
            Fin de si
            $vt_Balise_début:=""
        Fin de si

        Au cas ou
            : ($tt_Méthode_ligne1{$i}="@<NOM>@</NOM>")
                `On zappe, car le nom de la méthode est apporté par la macro

            : ($tt_Méthode_ligne1{$i}="@selection@")

            : (Non($vb_Insérer))

            Sinon
                $vl_Longueur:=Longueur($vt_Méthode_ligne1)
                $vl_Blob_taille:=Taille BLOB($1->)
                FIXER TAILLE BLOB($1->;$vl_Blob_taille+$vl_Longueur)
                Boucle ($j;0;$vl_Longueur-1)
                    $1->{$vl_Blob_taille+$j}:=Code ascii($vt_Méthode_ligne1≤$j+1≥)
                Fin de boucle
        Fin de cas

    Fin de boucle
Fin de si
`_____________________________________________________________________________________
`NETTOYAGE
Fin de si
`</EXE>
```

3. MTD_MethodList_Meth_CC4D_id_get

```
`<NOM>MTD_MethodList_Meth_CC4D_id_get</NOM>
`_____________________________________________________________________________________
`DESCRIPTION FONCTIONNELLE <DSC>
` This fonction checks and creates if necessary the directories for the module
` for which a code was given and return the working directory path
`Ex svn_MethodList_Meth_CC4D_id_get
`tt_SVN_methodList remplacé par $tt_Méthode
`tl_SVN_method_CC4D_ID remplacé par $tl_Méthode_CC4D
`</DSC>
`_____________________________________________________________________________________
`AMELIORATIONS A APPORTER <AML>
`</AML>
`_____________________________________________________________________________________
```

```
   `CREATION par Bruno LEGAY <DAT>
   `•1• 29/03/07, 21:06:56 - v1.00.00
   `•2• 30/04/07 10:24:17
   `</DAT>
   `_____________________________________________________________________________________
   `COMPOSANT <CMP>
   `</CMP>
   `_____________________________________________________________________________________
   `PARAMETRES<PRM>
C_ENTIER LONG($0)
C_TEXTE($1)
C_ENTIER($2)
   `PARAMETERS :
   `  $0 (TEXT)  <= working directory path
   `  $1 (TEXT) => structure path
   `  $2 (TEXT) => module code
   `</PRM>
   `_____________________________________________________________________________________
   `TYPAGE DES VARIABLES LOCALES <TVL>
C_BOOLEEN($vb_OK)
C_ENTIER LONG($vl_CC4D_id)
C_TEXTE($vt_Méthode)
C_ENTIER($ve_docRef)
C_ENTIER LONG($vl_NbParam)
C_BOOLEEN($vb_currentResFile)
   `</TVL>
   `TYPAGE DES VARIABLES à reclasser dans les COMPILER_XXX <TVG>
   `</TVG>
   `TYPAGE DES VARIABLES INTERPROCESS à reclasser <TVI>
   `</TVI>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES LOCALES <IVL>
$vb_OK:=Vrai
$vl_CC4D_id:=0
$vl_NbParam:=Nombre de parametres
   `</IVL>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES GLOBALES <IVG>
TABLEAU TEXTE($tt_Méthode;0)
TABLEAU ENTIER LONG($tl_Méthode_CC4D;0)
   `</IVG>
   `_____________________________________________________________________________________
   `CONTROLE DES PARAMETRES <CPR>
   `</CPR>
   `_____________________________________________________________________________________
   `EXECUTION
Si ($vb_OK)
      Si ($vl_NbParam>=1)    `>2) A RESTAURER
         $vt_Méthode:=$1

         Au cas ou
            : ($vl_NbParam=1)
               $ve_docRef:=-1
            Sinon
                  `: ($vl_NbParam=2)
               $ve_docRef:=$2
         Fin de cas

         $vb_currentResFile:=($ve_docRef=-1)    `(($vl_NbParam>2) | ($ve_docRef=-1))

         Si ($vb_currentResFile)
            MTD_MethodList_get (->$tt_Méthode;->$tl_Méthode_CC4D)
         Sinon
            MTD_MethodList_get (->$tt_Méthode;->$tl_Méthode_CC4D;$ve_docRef)
         Fin de si

         C_ENTIER LONG($vl_found)
```

```
            $vl_found:=Chercher dans tableau($tt_Méthode;$vt_Méthode)
            Si ($vl_found>=1)
                $vl_CC4D_id:=$tl_Méthode_CC4D{$vl_found}
            Sinon
                  `$vl_CC4D_id:=API_Resource_NewID`BE
                ALERTE("Méthode "+$1+" non trouvée")    `BE
            Fin de si

            TABLEAU TEXTE($tt_Méthode;0)
            TABLEAU ENTIER LONG($tl_Méthode_CC4D;0)

        Sinon
            ALERTE(Nom methode courante+" not enough parameters")
        Fin de si
        $0:=$vl_CC4D_id

Fin de si
```

4. MTD_MethodList_get

```
`<NOM>MTD_MethodList_get</NOM>
`_____________________________________________________________________________________
`DESCRIPTION FONCTIONNELLE <DSC>
` This method loads into arrays the method names and their CC4D ids for a given structure
`</DSC>
`_____________________________________________________________________________________
`AMELIORATIONS A APPORTER <AML>
`</AML>
`_____________________________________________________________________________________
`CREATION par Bruno LEGAY <DAT>
`•1• 29/08/06, 19:52:32 - v1.00.00
`•2• 08/05/07 10:25:57
`</DAT>
`_____________________________________________________________________________________
`COMPOSANT <CMP>
`</CMP>
`_____________________________________________________________________________________
`PARAMETRES<PRM>
C_POINTEUR($1)    `method names array pointer
C_POINTEUR($2)    `method IDs array pointer
C_ENTIER($3)    `4D structure file reference
C_ALPHA(4;$4)    `4D resource type
`  $1 (POINTER) => array text pointer for method names (object modified)
`  $2 (POINTER) => array longint pointer for method IDs (object modified)
`  $3 (INTEGER) => 4D structure file reference returned by svn__API_ResFile_Open
`   (optionnal, default current structure)
`  $4 (STRING) => 4D resource type (optionnal, default "TP4D")
`</PRM>
`_____________________________________________________________________________________
`TYPAGE DES VARIABLES LOCALES <TVL>
C_ALPHA(40;$va40_Méthode_courante)
C_ALPHA(4;$va4_Res_Type)    `We are checking that the 4D structure file reference looks ok (opened)
C_BLOB($vx_TP4D_Blob)
C_BOOLEEN($vb_currentResFile)
C_BOOLEEN($vb_fileOpenOk)
C_BOOLEEN($vb_OK)
C_ENTIER($ve_docRef)    `4D structure file reference
C_ENTIER LONG($i)
C_ENTIER LONG($vl_CC4D_ID)
C_ENTIER LONG($vl_CC4D_ID_Offset)
C_ENTIER LONG($vl_count)
C_ENTIER LONG($vl_errCode)
C_ENTIER LONG($vl_nbParam)
C_ENTIER LONG($vl_newSize)
C_ENTIER LONG($vl_offset)
C_ENTIER LONG($vl_recordLength)
C_ENTIER LONG($vl_resID)
C_ENTIER LONG($vl_start)
```

```
C_POINTEUR($vp_methodNameArrayPtr)    `method names array pointer
C_POINTEUR($vp_method_CC4D_ID_ArrayPtr)    `method IDs array pointer
C_TEXTE($vt_structPath)    `We are checking that the 4D structure file reference looks ok (opened)
TABLEAU ENTIER LONG($tl_method_CC4D_ID;0)
TABLEAU TEXTE($tt_methodName;0)
   `</TVL>
   `TYPAGE DES VARIABLES à reclasser dans les COMPILER_XXX <TVG>  `</TVG>
   `TYPAGE DES VARIABLES INTERPROCESS à reclasser <TVI>  `</TVI>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES LOCALES <IVL>
$vb_OK:=Vrai
$vl_nbParam:=Nombre de parametres
$vl_recordLength:=44
$vl_resID:=0
   `</IVL>
   `_____________________________________________________________________________________
   `INITIALISATION DES VARIABLES GLOBALES <IVG>
   `</IVG>
   `_____________________________________________________________________________________
   `CONTROLE DES PARAMETRES <CPR>
   `</CPR>
   `_____________________________________________________________________________________
   `EXECUTION <EXE>
Si ($vb_OK)
      Si ($vl_nbParam>1)    `We need at least two parameters

         $vp_methodNameArrayPtr:=$1
         $vp_method_CC4D_ID_ArrayPtr:=$2

            `Checking that the pointers are of the correct type
         Si ((Type($vp_methodNameArrayPtr->)=Est un tableau texte ) & (Type($vp_method_CC4D_ID_ArrayPtr-
      >)=Est un tableau entierlong ))

            `Handling default values for $3 and $4
         Au cas ou
            : ($vl_nbParam=2)
               $ve_docRef:=-1
               $va4_Res_Type:="TP4D"

            : ($vl_nbParam=3)
               $ve_docRef:=$3
               $va4_Res_Type:="TP4D"

            Sinon
                  `: ($vl_nbParam=4)
               $ve_docRef:=$3
               $va4_Res_Type:=$4
         Fin de cas
            `******************
            `svn__debug(Nom methode courante+" $ve_docRef = "+Chaine($ve_docRef))
            `******************
         $vb_currentResFile:=($ve_docRef=-1)

            `If ($vl_nbParam<3)    `If we did not provide a 4D structure file reference parameter,
            `    `we will have to open the current structure file ourselves
            `
            `svn__debug (Current method name+" opening ResFile = "+TXT_Enclose ($vt_structPath))
            `
            `    `Get the current structure path
            `C_TEXT($vt_structPath)
            `$vt_structPath:=Structure file
            `
            `    `and opening it...
            `$ve_docRef:=API_ResFile_Open ($vt_structPath)
            `End if

            `We are checking that the 4D structure file reference looks ok (opened)
```

```
$vb_fileOpenOk:=($ve_docRef#0)

Si ($vb_fileOpenOk)
	`**************
	`svn__debug(Nom methode courante+" calling API_Resource_Get with resType =
"+TXT_Enclose($va4_Res_Type)+", index = "+TXT_Enclose($va4_Res_Type)+" and docRef =
"+Chaine($ve_docRef))
	`******************
	`Get the list of methods and their IDs
	`$vl_errCode:=API Get Resource ($va4_Res_Type;0;$vx_TP4D_Blob;$ve_docRef)
	FIXER TAILLE BLOB(SVN_vx_TP4D_rsrc_Blob;0)

	Si ($vb_currentResFile)
		$vl_errCode:=API_Resource_Get ($va4_Res_Type;0;->SVN_vx_TP4D_rsrc_Blob)
	Sinon
		$vl_errCode:=API_Resource_Get ($va4_Res_Type;0;->SVN_vx_TP4D_rsrc_Blob;$ve_docRef)
	Fin de si

	$vx_TP4D_Blob:=SVN_vx_TP4D_rsrc_Blob
	FIXER TAILLE BLOB(SVN_vx_TP4D_rsrc_Blob;0)

	Si ($vl_errCode=0)

		`a method name record is composed of 44 bytes.
		`the first 32 bytes are the method name (31 character pascal string)
		`Then we have, from offset 32, two bytes for the CC4D ID (in native byte ordering)

		`Get the number of methods
		$vl_count:=Taille BLOB($vx_TP4D_Blob)\$vl_recordLength

		TABLEAU TEXTE($tt_methodName;$vl_count)
		TABLEAU ENTIER LONG($tl_method_CC4D_ID;$vl_count)

		$vl_start:=0
		Boucle ($i;1;$vl_count)	`Loop for each method name

			`Get the current method name
			$vl_offset:=$vl_start
			$va40_Méthode_courante:=BLOB vers texte($vx_TP4D_Blob;Chaîne pascal ;$vl_offset)

			`Get the CC4D ID
			$vl_CC4D_ID_Offset:=$vl_start+32
			$vl_CC4D_ID:=BLOB vers entier($vx_TP4D_Blob;Ordre octets natif ;$vl_CC4D_ID_Offset)

			$tt_methodName{$i}:=$va40_Méthode_courante
			$tl_method_CC4D_ID{$i}:=$vl_CC4D_ID

			$vl_start:=$vl_start+$vl_recordLength
		Fin de boucle

		`sorting the arrays (names and ids) by method name
		TRIER TABLEAU($tt_methodName;$tl_method_CC4D_ID;>)

		`copy the local arrays into the arrays given in the parameters
		COPIER TABLEAU($tt_methodName;$vp_methodNameArrayPtr->)
		COPIER TABLEAU($tl_method_CC4D_ID;$vp_method_CC4D_ID_ArrayPtr->)

		`free ou local arrays
		TABLEAU TEXTE($tt_methodName;0)
		TABLEAU ENTIER LONG($tl_method_CC4D_ID;0)

	Sinon
		ALERTE(Nom methode courante+" error "+Chaine($vl_errCode)+" calling API Get Resource
\""+$va4_Res_Type+"\" !")
	Fin de si

	`If (($vl_nbParam<3) & $vb_fileOpenOk)	`We did open the file so we will close it
```

```
                    `API_ResFile_Close ($ve_docRef)
                    `End if

                Sinon     `$va4_Res_Type = 0, there is a problem with the file reference
                    ALERTE(Nom methode courante+" error "+Chaine($vl_errCode)+" opening resource file !")
                Fin de si

            Sinon     `not of the expected type
                ALERTE(Nom methode courante+" error with $1 and/or $2 paramters !")
            Fin de si

        Sinon
            ALERTE(Nom methode courante+" : not enough parameters.")
        Fin de si


        `_____________________________________________________________________________________________
        `NETTOYAGE
Fin de si
    `</EXE>
```

5. API_Resource_Get

```
    `<NOM>API_Resource_Get</NOM>
    `_____________________________________________________________________________________________
    `DESCRIPTION FONCTIONNELLE<DSC>
    ` This function is a wrapper on "API Get Resource"
    `</DSC>
    `_____________________________________________________________________________________________
    `AMELIORATIONS A APPORTER <AML>
    `</AML>
    `_____________________________________________________________________________________________
    `CREATION par Bruno LEGAY (BLE)<DAT>
    `•1• 29/03/07, 08:19:59
    `•2• 28/09/2007 00:24:12 <TAG>
    `</DAT>
    `_____________________________________________________________________________________________
    `COMPOSANT <CMP>
    `</CMP>
    `_____________________________________________________________________________________________
    `PARAMETRES <PRM>
C_ALPHA(4;$1)    `$1 (STRING 4) => resource type
C_ENTIER LONG($2)    `$2 (LONGINT) => resource id
C_POINTEUR($3)    `$3 (POINTEUR) => pointer to a blob (modified)
C_ENTIER($4)    `$4 (INTEGER) => 4D structure file reference (returned by "API_ResFile_Open")
    `            optionnal, default, the current structure
C_ENTIER LONG($0)    `$0 (LONGINT)  <= error code (0 if everything went fine)
    `</PRM>
    `_____________________________________________________________________________________________
    `TYPAGE DES VARIABLES LOCALES <TVL>
C_BOOLEEN($vb_OK)
C_ENTIER LONG($vl_errCode)    `Err Code
C_ALPHA(4;$va_resType)    `Resource type
C_ENTIER LONG($vl_resId)    `Resource id
C_POINTEUR($vp_resBlobPtr)    `Pointer to a blob (modified)
C_ENTIER($ve_docRef)    `4D structure file reference
C_ENTIER LONG($vl_nbParam)
    `</TVL>
    `TYPAGE DES VARIABLES à reclasser dans les COMPILER_XXX <TVG>
    `</TVG>
    `TYPAGE DES VARIABLES INTERPROCESS à reclasser <TVI>
    `</TVI>
    `_____________________________________________________________________________________________
    `INITIALISATION DES VARIABLES LOCALES <IVL>
$vb_OK:=Vrai
    `</IVL>
    `_____________________________________________________________________________________________
```

```
   `INITIALISATION DES VARIABLES GLOBALES <IVG>
   `</IVG>
   `_____________________________________________________________________________________
   `CONTROLE DES PARAMETRES <CPR>
$vl_nbParam:=Nombre de parametres
  `</CPR>
   `_____________________________________________________________________________________
   `EXECUTION <EXE>
Si ($vb_OK)
        $vl_errCode:=0

        Si ($vl_nbParam>2)
            $va_resType:=$1
            $vl_resId:=$2
            $vp_resBlobPtr:=$3

            Au cas ou
                : ($vl_nbParam=3)
                        `$ve_docRef:=
                Sinon
                    $ve_docRef:=$4
            Fin de cas

            Si (Type($vp_resBlobPtr->)=Est un BLOB )
                    `Using API_Pack
                    `We are using a local blob to get API Get Resource to work

                    `Using a deferenced blob pointer does not work (Bruno LEGAY 30/03/2007)
                    `$vl_errCode:=API Get Resource ($va_resType;$vl_resId;$vp_resBlobPtr->;$ve_docRef)
                    `****************
                    `svn__debug(Nom methode courante+" - getting resource - type : "+TXT_Enclose($va_resType)+" - id
        : "+Chaine($vl_resId)+" - blob : "+Chaine(Taille BLOB($vx_tempBlob))+" - docRef : "+Chaine($ve_docRef))
                    `****************
                C_BLOB($vx_tempBlob)
                FIXER TAILLE BLOB($vx_tempBlob;0)

                Au cas ou
                    : ($vl_nbParam=3)
                        $vl_errCode:=API Get Resource ($va_resType;$vl_resId;$vx_tempBlob)

                    : ($ve_docRef=-1)    `This is the current structure
                        $vl_errCode:=API Get Resource ($va_resType;$vl_resId;$vx_tempBlob)

                    Sinon
                        $vl_errCode:=API Get Resource ($va_resType;$vl_resId;$vx_tempBlob;$ve_docRef)

                Fin de cas
                    `**********************
                    `svn__debug(Nom methode courante+" - getting resource - blob : "+Chaine(Taille
        BLOB($vx_tempBlob))+" byte(s) - errCode : "+Chaine($vl_errCode))
                    `********************
                $vp_resBlobPtr->:=$vx_tempBlob

                FIXER TAILLE BLOB($vx_tempBlob;0)

            Sinon
                ALERTE(Nom methode courante+" : $3 is not a pointer to a blob")
            Fin de si

        Sinon
            ALERTE(Nom methode courante+" : not enough parameter")
        Fin de si
        $0:=$vl_errCode
Fin de si
   `</EXE>
```