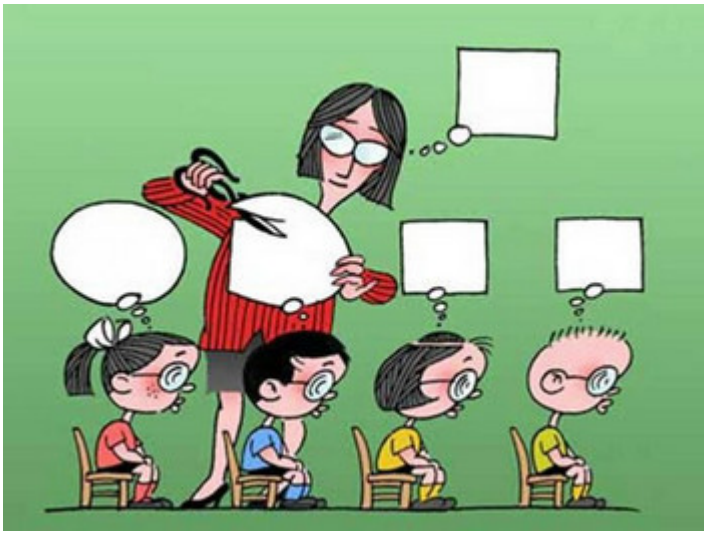
Normes de développement Gestan *



Normaliser, c'est empiéter sur la créativité naturelle des auteurs, mais cela permet de faciliter le travail des développeurs quand plusieurs personnes interviennent sur un programme, et de ne pas déconcerter les utilisateurs par des ergonomies ou des apparences hétérogènes.

Aucune norme ne représente un idéal, mais seulement un choix subjectifs parmi tous les choix possibles.

Si elles sont les plus légères possibles, ces normes doivent être respectées au plus près.

Normes communes

Gabarit

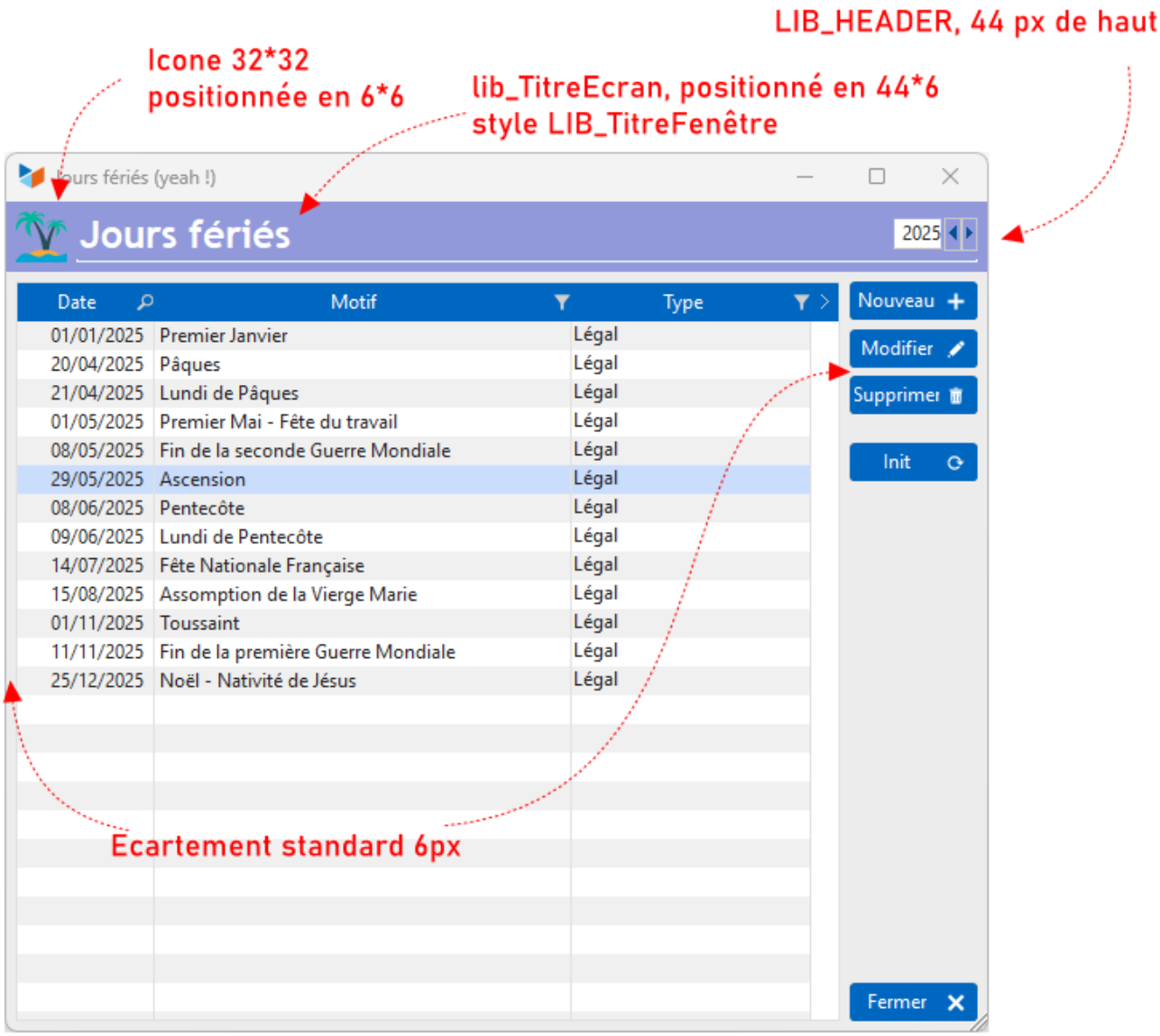
- Utilisez le gabarit GestanGB (Gris-Bleu).
- Utilisez les icônes flat design, incluses dans le set d'icônes Gestan.

Vous trouverez ces éléments sur la page ressources.

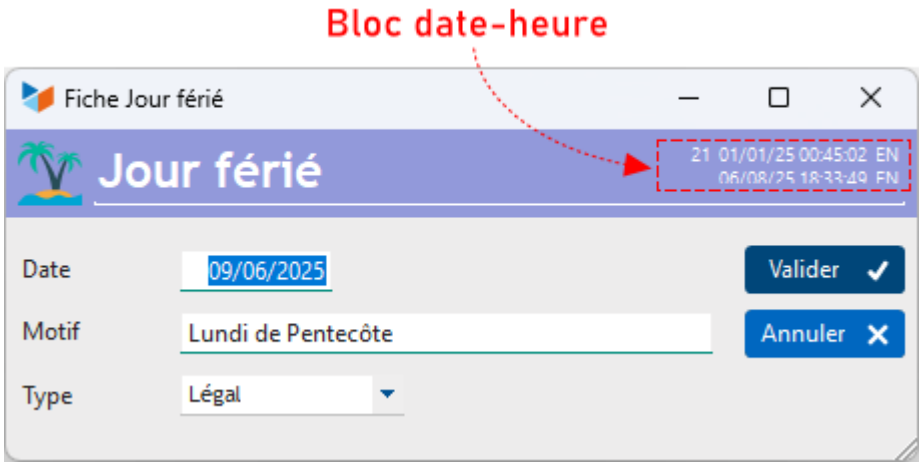
La police des menus est Segoe UI, de couleur gris 546368, de couleur bleu clair au survol E8F0FE

Ecrans

Table standard



Fiche standard



C'est désormais une fonction de Gestan qui alimente le bloc date/heure du coin haut droite : normalisation facile par copier-coller. 😉

```
lib_DateHeure = gf_Alimenter_DateHeure("JRFERIE")
```



Charte de programmation de Windev

Son utilisation est à éviter (alourdissement non indispensable), des conventions de nommage plus légères sont suffisantes, voir ci-dessous.

Traductions

Gestan est actuellement traduit en espagnol (langue 1) et en anglais (langue 2) : il faut intégrer ces deux langues dans les langues du projet.

Pour les libellés devant être traduits, il faut générer la pré-traduction avec un Ctrl+T sur chaque libellé concerné.

Pour faciliter la traduction, tous les libellés composés dynamiquement doivent utiliser la fonction ChaineConstruit, par exemple :

```
// A ne pas faire
Info("Le contact " + CONTACT.NOMFAMILLE + " est inconnu.")
// A faire
Info(ChaineConstruit("Le contact %1 est inconnu.", CONTACT.NOMFAMILLE))
```

Fichiers

Tous les fichiers doivent posséder :

- un ID automatique sur 8 bits, nommée ID[nom du fichier], par exemple IDZ27_TEST pour le fichier Z27_TEST
- les 4 rubriques suivantes : DHCRE et DHMOD (dateheure de création et de dernière modification), USCRE et USMOD (code utilisateur créateur et dernier modifieur). Ce sont les timestamp de création et de modification, un trigger dans Gestan les met à jour automatiquement.

Conventions de nommage

Fichiers

- Les noms de fichier sont toujours en majuscules. C'est plus clair.
- Les fichiers de données des extensions sont préfixés par Zxx_, par exemple Z43_PROSPECT (xx étant le numéro d'extension attribué par ICS). Cela présente pour intérêt de pouvoir identifier facilement les fichiers utilisés par les bibliothèques, et de garantir que votre fichier ne soit pas écrasé par un autre de même nom d'une autre extension (voir la liste).
- Les noms de fichiers de liens commencent par LK_ (Donc Z99_LK_ pour un fichier de lien utilisé par une extension).

Rubriques des fichiers

- Les noms de rubrique sont toujours en majuscules. C'est plus clair.
- Les noms de rubrique sont préfixés ainsi : les dates par DT_, les taux par TX_, les montants par MTT_, les libellés par LIB_, les nombres par NB_, les images par IMG_, les types par TYP_. Quand il s'agit d'un ID se rapportant au fichier des libellés paramétrables PARAMLIB, le nom de la rubrique est préfixé par ID2_ (par exemple ECRITURE.ID2_MODEPAIEMENT)

Variables

- le préfixe g indique une variable globale de Gestan “core”, éviter de l'utiliser dans vos programmes. Par exemple, gIniFile pour le fichier de paramétrage de Gestan, gOptFile pour le fichier de paramétrage de la base en cours, etc...
- le préfixe b indique un booleen
- idx : index à tout faire, par exemple pour parcourir un fichier ou une table.
- SCR : chaine à tout faire (variable de travail)
- le préfixe w sur une variable indique une variable de travail

Écrans

- les tables des écrans liste sont préfixées par tbl_, puis le nom du fichier concerné, par exemple tbl_CONTACT (pour éviter les confusions avec les noms des écrans, par exemple Table_CONTACT)
- vous pouvez, sans obligation, préfixer les boutons par btn_, les interrupteurs par int_, les sélecteurs par sel_, les combos par cbo_.
- les noms des fonctions dans les programmes sont structurées par le verbe à l'infinitif et le complément, par exemple la fonction “Alimenter_Table” pour l'alimentation d'une table mémoire, plutôt que “TableAlimente”, ou “ProcAlimTbl”.

Images

Dans la mesure du possible, n'utilisez que les images présentes dans le set d'icônes de Gestan.

Si vous utilisez une image propre à votre extension :

- nommez-la selon cette convention : [extension]_nomimage_[couleur]_[taille]_[nb états].png, avec [extension]=préfixe de votre extension, [couleur]=couleur de la variante de couleur éventuelle. Par exemple z27_thunder_64_1.png, ou z27_bullet_red_64_1.png. L'usage du préfixe évite que votre image soit remplacée par une image de Gestan core.

Format des zones numériques

Zones représentant des montants monétaires

Le nombre de décimales des montants monétaires est paramétrable. Pour prendre ce paramètre en compte et mettre les montants au bon format, utilisez la fonction gf_Renvoyer_MasqueNum().

Par exemple, dans le code d'initialisation de l'écran ou de l'état :

```
MTTHT_Crédit..MasqueSaisie  = gf_Renvoyer_MasqueNum()
MTTHT_Débit..MasqueSaisie   = MTTHT_Crédit..MasqueSaisie
```

Pour le mode test, vous pouvez faire :

```
MTTHT_Crédit..MasqueSaisie  = EnModeTest() ? "999 999,99" SINON
gf_Renvoyer_MasqueNum()
MTTHT_Débit..MasqueSaisie   = MTTHT_Crédit..MasqueSaisie
```

Le nombre de décimales des prix d'achat reste libre (notamment pour tenir compte des prix d'achats au mille d'éléments devant être revendus).

Zones représentant des quantités de produit

Le nombre de décimales des quantités de produit est paramétrable dans Gestan. Pour prendre ce paramètre en compte et mettre les quantités de produit au bon format, utilisez la fonction gf_Renvoyer_MasqueNum()avec le paramètre “QTPROD”.

Par exemple :

```
QTE_PRODUIT..MasqueSaisie   = gf_Renvoyer_MasqueNum("QTPROD")
```

Normes de développement pour les programmes

Commentaires

Commentez vos programmes !

Une proportion de 20% de commentaire par rapport aux lignes de code est un minimum.

N'écrivez pas pour ne rien dire, comme par exemple :

```
// Fermeture de l'écran
Ferme()
```

mais écrivez pour expliquer ce que fait votre programme et pourquoi.

Fonctions normalisées :

Fonction de gestion des erreurs, à mettre dans le code d'initialisation de la fenêtre.

A noter que la fonction gf_InfoErreur est utilisable dans beaucoup de cas d'erreurs. Le premier paramètre est le type d'erreur : c'est une variable globale commençant par MSG_

```
// Gestion des erreurs d'accès à la base de données
SI PAS EnModeTest() ALORS
    QUAND EXCEPTION
        gf_InfoErreur(MSG_ERRINATTENDUE,2,ExceptionInfo(errMessage))
        ExceptionActive()
        RepriseSaisie()
    FIN
FIN
```

Coloration des zones de l'écran

A mettre dans l'init

```
gf_Coloriser()
```

** Alimentation de la zone “timestamp”**

A mettre juste après le FichierVersEcran() qui est habituellement en fin d'initialisation de la fiche. Le libellé TimeStamp doit s'appeler lib_DateHeure.

```
lib_DateHeure = gf_Alimenter_DateHeure("JRFERIE")
```

Fonction de contrôle de lecture : gf_InfoPasTrouve. Cette fonction affiche un message standard pour enregistrement non trouvé.

```
SI PAS HLitRecherche(CONTACT,IDCONTACT,VALKLE) ALORS
gf_InfoPasTrouve("CONTACT",VALKLE);RETOUR
```

Fonction de suppression d'enregistrement :

Ici, un exemple de suppression dans la table des absences.

```
SI TableSelect(tbl_ABSENCE)=-1 ALORS RETOUR
// Contrôles avant validation (éventuellement)
SI ABSENCE.IVALIDE ALORS Info("Cette absence est validée, suppression non
autorisée.");RETOUR

SI PAS gf_Info_Erreur(MSG_SUPPRIMER,3) ALORS RETOUR
TableSupprime(tbl_ABSENCE)
TableAffiche(tbl_ABSENCE, taCourantPremier)
```

Fonction d'exécution d'un traitement batch : gf_Info_Traitement. Cette fonction affiche un message standard à la fin d'une exécution, par exemple la mise à jour d'un fichier. Par exemple :

```
gf_Info_Traitement(nbOK,"M",nbKO,nbParcours,"Libellé complémentaire
éventuel")
```

affichera : “Opération terminée OK, 10 enregistrements parcourus, 7 enregistrement modifiés, 0 erreurs détectées, Libellé complémentaire éventuel.”

Aide de vos programmes

La présence d'une documentation est obligatoire. Elle doit être disponible dans les trois langues, et accessible via un bouton de type F1, avec le code suivant :

```
gf_Aide("https://manuel.gestan.fr/doku.php?id=fr:wiki:referentiel:joursferie
s")
```

Ce code va ouvrir la page d'aide dans la bonne langue (fr, es, ou en), en fonction de la lange de l'interface.

Normes de développement pour les écrans

Utilisez les fonctions de positionnement automatique de Windev pour normaliser les écarts entre les zones.

Utilisez la feuille de style Gestan pour les champs, les tables, etc.

Le bouton “Fermer” est toujours en bas à droite des écrans liste.

Normes de développement pour les états

- Le nom de l'état est Etat_NNN99_Description, par exemple Etat_ACT2_fiche_action, dans lequel

NNN est une abréviation du fichier principal concerné, et 99 un numéro d'ordre.

- Les marges sont à 15 mm sur tous les côtés.
- La police est Trébuchet MS (ancienne norme), ou Segoe UE (nouvelle norme).
- Le bloc début de document fait 16 mm de haut pour un état portrait, 12 mm pour un état paysage.
- La hauteur des têtes de colonnes est de 6 mm. La couleur de fond des têtes de colonne est paramétrable. Tous les libellés d'entête de colonne sont dans le groupe GP_ENTETE.
- Le début de document contient le logo de la société, en général le logo réduit, dans une image appelée IMG_LOGO
- La hauteur d'une ligne de corps est de 5 ou 6 mm selon les cas.
- Les couleurs de fond sont normalisées, ce sont des variables globales préfixées par COUL_
- La hauteur du bas de page est de 5 mm.
- Le bas de page est normalisé : il contient le nom de l'état, la date et l'heure d'impression, le code utilisateur demandant l'impression, le libellé bas d'états paramétrable (dans le champ LIB_BASETAT), et l'indication de la pagination (page n/x)

Il faut activer les masques d'affichage des monétaires et des quantités de produit, en utilisant les fonctions ad hoc, comme par exemple, dans la fonction d'initialisation de l'état :

```
MONTANT_1..MasqueSaisie = gf_Renvoyer_MasqueNum()
MONTANT_2..MasqueSaisie = MONTANT_1..MasqueSaisie
```

Vous pouvez mettre dans la fonction d'initialisation de l'état les lignes suivantes :

```
// Initialisations de l'état
HLitPremier(PARAMAPPLI,RAISON_SOCIALE)
IMG_LOGO = PARAMAPPLI.LOGO_R
LIB_BASETAT = PARAMAPPLI.LIB_BAS_ETAT
GP_ENTETE..CouleurFond = PARAMAPPLI.CL_FONDCOL
LIB_IDETAT=ExtraitChaîne(MonEtat..Nom,2,"_")+" -
"+DateVersChaîne(DateSys(),"JJ/MM/AA")+"
"+HeureVersChaîne(HeureSys(),"HH:MM")+" - "+gCdUser
```

Autres articles “Développement”

Développement d'extensions en version Ax
Intégrer une pièce dans Gestan depuis une extension
Ressources
Rédiger des pages d'aide sur le wiki

From:
https://manuel.gestan.fr/ - **Le manuel de Gestan**

Permanent link:
https://manuel.gestan.fr/fr/partenaires/normesdev?rev=1754499248

Last update: **2025/08/06 16:54**