



MF category **Comment paramétrer le "Thème0" - Appliquer sa charte graphique**

Ce guide explique comment paramétrer le "Thème0" de FormPublisher v2.1 pour ajuster les interfaces web [HTML](#) des applications produites :

- image bannière
- couleurs
- polices de caractères
- pictogrammes
- boutons
- labels
- navigation

Prérequis : connaissance de CSS (propriétés, sélecteurs, règles, priorités/surcharges)

1. Ressources ergonomiques et thèmes

FormPublisher est livré avec un ensemble de ressources de référence pour produire les interfaces web : des [templates](#), des pictogrammes, des CSS, etc. Cet ensemble est appelé le "ref. standard".

Un "thème" (déclinaison ergonomique) est adjoint au "ref standard" pour publier les applications. Ainsi, une même application peut être publiée avec différents "thèmes" ; les publications seront isofonctionnelles et visuellement distinctes. Par exemple une application avec deux "visages" : extranet et intranet.

FormPublisher est livré de base avec un "thème", le "Theme0" aussi appelé thème "standard".

Pour modifier une des ressources ergonomiques utilisée par la publication, il suffit de la surcharger dans un dossier APPLI.REF spécifique à l'application. Les chapitres suivants énumèrent la liste des ressources du "ref. standard" et du "Theme0" qui peuvent être ainsi surchargées via APPLI.REF et montre comment le réaliser.

2. Surcharge APPLI.REF

Dans le dossier APPLI.REF de la publication, faire les sous-dossiers suivants :

- css
- images
- [xslt](#)

C'est dans ces sous-dossiers que sont placés les ressources qui prennent le pas sur celles du "ref. standard" et du "Theme0". Ces ressources sont :

- APPLI.REF/css/**theme.css** (uniquement cette CSS)
- APPLI.REF/images/actions/*.png (11 png)
- APPLI.REF/images/background/banner.png
- APPLI.REF/images/icons/active_title.png
- APPLI.REF/images/icons/back2top.gif
- APPLI.REF/images/list/bullet*.gif (11 gif)
- APPLI.REF/images/list/bullet*.png (5 png)
- APPLI.REF/images/logo/**logo2.png**
- APPLI.REF/images/map/level*.gif (9 gif)
- APPLI.REF/images/menu/border_bottom.gif
- APPLI.REF/images/menu/border_top.gif
- APPLI.REF/images/menu/level*.gif (4 gif)
- APPLI.REF/images/menu/selected.gif
- APPLI.REF/images/panel/panel*.jpg (4 jpg)
- APPLI.REF/images/step/button.png
- APPLI.REF/images/step/next.png
- APPLI.REF/images/step/previous.png
- APPLI.REF/xslt/**i18nlabels.xml** (1 file for all customised labels)

Les chapitres suivants exposent chaque élément, notamment les rôles des sélecteurs dans /ref/css/theme.css.

3. Bannière d'entête

La **bannière** est une **image unique**.



Placer l'**image** à substituer à la bannière dans APPLI.REF\images\background**banner.jpg**

Avec **height** ajuster la hauteur de la bannière à la taille de la nouvelle image dans APPLI.REF\css**theme.css** :

```
#header { height: 96px; }
```

4. Liséré sous bannière

Un **liséré** gris souligne la bannière. Il affiche le MetaField **category** et le MetaField **application**.

Pour changer le style du liséré, voir APPLI.REF\css\theme.css :

```
#banner_title {
background-color: #00AA00;
border-top: 1px solid red;
}
```

Il est bien sûr possible de changer d'autres propriétés CSS, comme **height** par exemple.

Les MetaField s'inscrivent dans des . Pour changer leur style, voir APPLI.REF\css\theme.css :

```
#banner_title span { color: #FFFF00; }
```

Enfin, il est possible de changer le style de chaque MetaField séparément, car chacun porte son nom comme classe CSS :

- #banner_title span.**category** { color: purple; }
- #banner_title span.**application** { font-size: 30px; }

5. Textes d'interface (labels)

Les **labels** (textes d'interface) tels "Aide", "Aperçu PDF", "Page suivante", "Pourcentage supérieur à 100", etc sont issus du fichier REF\xslt\i18nlabels.xml. Ce fichier contient une section par langue (lb, de, en, fr).

Pour **modifier un label**, il suffit de placer dans APPLI.REF\xslt un fichier i18nlabels.xml sur ce [modèle](#) :

```
<?xml version="1.0" encoding="UTF-8"?>
<labels>
  <fr>
    <labelValidate>Soumettre</labelValidate>
    <labelErrorPercentage>Le pourcentage saisi est supérieur à 100.</labelErrorPercentage>
  </fr>
</labels>
```

Comme pour les CSS, on peut se contenter de ne surcharger que le nécessaire (seulement 2 labels pour une seule langue dans cet exemple).

⚠ Attention : veillez à bien [enregistrer](#) le fichier **i18nlabels.xml** en **UTF8**. L'[enregistrement](#) avec un outil Windows tel Notepad peut produire des caractères Windows/ANSI, donc non UTF8 et rendre le fichier inutilisable.

Le nom de chaque label (ici labelValidate et labelErrorPercentage) est décrit ci-après dans son contexte.

6. Bas d'écran

Un **bandeau** est affiché au **bas de l'écran**, par exemple :

👉 Publié avec FormPublisher v2.1 | Version 42 Mise à jour le: 04-10-2011 | Copyright J-Way

ainsi qu'à droite un lien "Haut de page".

Contenus ensemble par un <div> **#footer**, chaque élément est un portant une classe :

- poweredBy
- version
- update
- copyright
- backtotop

Les barres verticales de séparation ("|") sont aussi dans des de classe "sep".

Les **labels** utilisés sont :

- labelFooterPoweredBy : Publié avec FormPublisher
- labelFooterVersion : Version
- labelFooterUpdate : Mise à jour le
- labelFooterCopyright : Copyright
- labelFooterBackToTop : Haut de page

La version de FormPublisher est indiquée automatiquement. La "Version" de la publication est la valeur de **jway.publication.version** dans APPLI.REF\publication.properties ou 1 par défaut. La date de mise à jour est la date de publication. Le copyright est attribué au MetaField "name" des MetaFields "[organisation](#)" des "Resources" du document.

Pour modifier un des labels, le **surcharger** dans APPLI.REF\xslt\i18nlabels.xml

Pour modifier un style, **ajouter** des directives CSS dans APPLI.REF\css\theme.css

Exemple :

```
👉 #footer span.update{
  background-color: green;
  font-size: 2em;
}
```

7. Onglets, vues et boutons

Les documents statiques (Nature=classic) affichent un **bouton** "PDF".

Les **formulaires** (Nature=form) affichent :

- une barre de boutons, dont "PDF", "Quitter"...
- des onglets pour basculer entre 2 "vues" : renseignements et [formulaire](#)

Pour changer les **pictogrammes** utilisés pour les **boutons**, placer les images de substitution dans APPLI.REF\images\actions sous les noms help.png, pdf.png, load.png, save.png, quit.png, contact.png, report.png.

Les versions désactivées sont nohelp.png, noload.png, nosave.png et noquit.png.

Pour changer les **pictogrammes** utilisés pour les **onglets**, placer les images de substitution dans APPLI.REF\images\panel sous les noms panelForm.jpg, panelForm_selected.jpg, panelInfo.jpg, panelInfo_selected.jpg.

Pour styler le **texte sur les onglets**, utiliser APPLI.REF\css\theme.css :

```
#menuPanel input.panel {color: purple;}
#menuPanel input.panelForm.selected {color: purple;}
#menuPanel input.panelForm {font-style: italic;}
#menuPanel input.panel.selected {font-style: italic;}
```

Le **bouton d'aide** est activé par les MetaFields du groupe "form" :

- soit une [URL](#) dans le MetaField "link-to-helpdesk" (pour pointer un site préexistant)
- soit un DocumentId dans le MetaField "[instructions](#)" (pour pointer un document de la publication)

Dans les formulaires, le **clik sur PDF** dépend de la vue :

- dans la vue "renseignements", le bouton PDF fournit le formulaire vide (formulaire "papier")
- dans la vue "formulaire", le bouton PDF fournit le formulaire avec les informations saisies en cours d'interview

8. Zone principale

La zone principale est comprise sous l'entête, les boutons et les éventuels onglets. Elle comprend une colonne map/menu à gauche et le contenu à droite, avec les boutons de navigation previous/next.

Le sélecteur CSS varie avec la vue :

- document statique : #mainzone
- [formulaire](#) : #mainzoneForm
- vue "renseignements" : #mainzoneInfo

Pour encadrer la zone principale, utiliser border: taille solid couleur

Pour juste placer un liséré horizontal en haut et en bas, ajouter border-right: none; et border-left: none;

```
#mainzone {
  background-color: lightgreen;
  border: 11px solid yellow;
  border-right:none;
  border-left:none;
}
#mainzoneInfo {
  background-color: lightSeaGreen;
  border: 11px dotted brown;
}
#mainzoneForm {
  background-color: lightgreen;
  border: 11px double yellow;
}
```

9. Titres

et sous-titres non affichés

Titres et **sous-titres** sont aussi stylés par APPLI.REF\css\theme.css pour changer couleurs, polices, etc.

Dans un [formulaire](#), le **titre** du document est stylé par :

```
div#mainzoneForm h2 {
  background-color: green;
  color: yellow;
}
```

le **Title** du document est remplacé par le ShortTitle s'il est présent.

Dans un formulaire, les sous-titres de document NE sont PAS affichés dans l'interface [HTML](#).

Dans un formulaire, le **titre** de section courante est stylé par :

```
👉 div#mainzoneForm div#form_edition h3 {  
  background-color: orange;  
}
```

Dans un formulaire, le **sous-titre** de section courante est stylé par :

```
👉 #mainzoneForm div.SectionSubtitle {  
  background-color: brown;  
  color: white;  
}
```

Dans un **document statique**, le **titre** de document est stylé par :

```
👉 #mainzone h2 {  
  color: red;  
  font-size: 1.5em;  
}
```

Dans un document statique, le **titre** de la section courante est stylé par :

```
👉 #mainzone h3 {  
  color: red;  
  font-size: 1.5em;  
}
```

Dans un document statique, les **sous-titres** de document et de section courante NE sont PAS affichés dans l'interface HTML.

Dans la vue "renseignements" des [formulaires](#), le style du **titre** du MetaContent courant est modifié par :

```
👉 #mainzoneInfo #edition h3.active_title {  
  background-color: pink;  
}
```

Dans la vue "renseignements" des [formulaires](#), le style du **sous-titre** du MetaContent courant est modifié par :

```
👉 #mainzoneInfo XYZ.active_stitle{  
  background-color: purple;  
}
```

10. Map et menu (colonne gauche)

La "**map**" des [formulaires](#), le "**menu**" des documents classiques (menu.xml) ainsi que la liste des MetaContent en vue "renseignements" s'affichent dans une **colonne à gauche** dans la zone principale.

Haut et bas de la colonne de gauche sont dessinés par

- APPLI.REF/images/menu/border_top.gif
- APPLI.REF/images/menu/border_bottom.gif

Entre les deux sont utilisés menu/level et list/bullet :

- APPLI.REF/images/list/bullet*.gif (11 gif) : bullet1.gif à bullet5.gif et bullet1H.gif à bullet5H.gif.

Enfin, l'élément en cours est APPLI.REF/images/menu/selected.gif

11. Vue renseignements

Dans la vue renseignements, la "map" de navigation à gauche est remplacée par la liste des "**MetaContent**" des "Resources" du [formulaire](#), tandis que la zone principale de contenu affiche le "MetaContent" sélectionné.

Lorsque l'auteur pose un **Title** sur un MetaContent, celui-ci se substitue au **label** du MetaContent dans la zone principale de contenu, le label restant utilisé dans la colonne de gauche.

Les **labels des MetaContent** sont :

- labelInfoDescription : MC "presentation"
- labelInfoAdvantages : Avantages
- labelInfoAppeal : Recours
- labelInfoSpecificAppeal : Recours spécifique
- labelInfoConditions : Conditions
- labelInfoContact : Contacts
- labelInfoHelp : Aide
- labelInfoInstructions : Instructions
- labelInfoGlossary : [Glossaire](#)
- labelInfoFaq : FAQ
- labelInfoLegal : Contexte légal et directives
- labelInfoPrivacy : Vie privée
- labelInfoProcess : Traitement
- labelInfoAdminProcess : Procédure administrative
- labelInfoSubject : Objet
- labelInfoTarget : Cible
- labelInfoMediator : Médiateur
- labelInfoOrganisation : Coordonnées de l'[organisation](#)
- labelInfoPhone : Tél
- labelInfoFax : Fax
- labelInfoContact : Contacts
- labelInfoLanguages : Langues
- labelInfoAbout : A propos
- labelInfoVersion : Version

Le MetaContent "**contact**" affiche labelInfoOrganisation s'il n'a pas de titre, puis son contenu et enfin les **MetaFields** suivants :

- name
- department1
- department2
- department3
- department4
- po-box
- address-line1
- address-line2
- zip-code
- city
- phone
- fax
- website
- email
- identifieur1
- identifieur2
- identifieur3
- identifieur4

Enfin, le MetaContent "contact" affiche l'image REF\images\logo\logo2.png de 137x80 pixels.

12. Contenus, questions et saisies

Les "Content" et les "Table" ont un StyleName, les "List" un "markerType". Ils portent des classes CSS permettant de les identifier et de les styler, ces styles CSS sont redéfinissables dans APPLI.REF\css\theme.css.

Les indications CSS suivantes sont indépendantes du placement des "Question" dans un "QuestionSet" ou directement dans un "Paragraph".

Pour modifier le style du **titre** d'un QuestionSet, surcharger APPLI.REF\css\theme.css :

 #form_edition fieldset.QuestionSet legend { background-color: orange; color: #64A004; }

Pour modifier le style des **champs de saisie**, surcharger APPLI.REF\css\theme.css :

 input.TextBox { background-color: lightBlue; }
input.PasswordBox { background-color: orange; }
select.ListBox { border: 4px dashed aqua; }
select.ComboBox { border: 4px solid aqua; }
textarea.MemoBox { color: darkblue; font-family: cursive; }

Existent aussi input.**CheckBox**, input.RadioBoxOption et span.**RadioBoxOption**.

Enfin, une classe CSS "**error**" ou "**warning**" est ajoutée pour permettre des sélections telles input.TextBox.**error** {color: red;} et input.TextBox.**warning** {color: orange;}

Les **pictogrammes** des contrôles sur champ de saisie sont à surcharger dans APPLI.REF\images\icons. Lorsqu'un

champ de saisie est obligatoire, il est précédé de  fieldRequired.png. Lorsqu'il est en avertissement, il est précédé de

 fieldWarning.png. Lorsqu'il est en erreur, il est précédé de  fieldError.png. Enfin, fieldNoMark.png sert pour les autres champs.

Les contrôles explicitent leurs messages via des **labels** :

- labelErrorRequiredField : Ce champ est obligatoire
- labelErrorPercentage : Pourcentage supérieur à 100
- labelErrorNumber : Il faut un nombre dans ce champ

(Voir les fichiers i18nlabels.xml du REF standard pour la liste complète)

Les pages comportant des questions avec **contrôles de saisie** peuvent afficher en **haut de page** un bandeau d'**avertissement** ou de signalement d'**erreur(s)** pour la page en cours. Pour modifier le style de ce bandeau, adapter APPLI.REF\css\theme.css :

 #form_edition div.errorOnPage ul.errorList { border: 3px dashed red; }
#form_edition div.errorOnPage ul.warningList { border: 3px dashed orange; }

Les labels avertissement/erreur en haut de page sont :

- labelGenericWarning : Des avertissements sont présents, veuillez les vérifier
- labelGenericError : Des erreurs sont présentes, veuillez les corriger

A partir de la version 2.0.1, FormPublisher offre la possibilité de distinguer le libellé s'il y a une seule erreur ou un seul warning (améliore l'accessibilité du **formulaire** aux mal-voyants)

- labelGeneric**Single**Warning : Un avertissement est présent, veuillez le vérifier
- labelGeneric**Single**Error : Une erreur est présente, veuillez la corriger

13. Boutons previous/next

Dans les **formulaires** dynamiques les "step" **boutons** d'interview (avant/arrière) sont redéfinis par APPLI.REF\css\theme.css :

 ○ #mainzoneForm div.step input.**previous** { border: 4px groove blue; }
○ #mainzoneForm div.step input.**next** { border: 4px groove red; }

Existe aussi une classe "input.disabled" :

 #mainzoneForm div.step input.**disabled** { border: 4px dashed darkgray; }

Bien sûr, là encore, d'autres styles CSS peuvent être appliqués à volonté.

Les textes de ces boutons sont définis par :

- labelPreviousStep : Page précédente
- labelNextStep : Page suivante

Lorsqu'on survole un **bouton** avec la souris, une bulle s'affiche avec ces labels :

- labelPreviousStepTitle : Retour à la page précédente
- labelNextStepTitle : Aller à la page suivante

L'effet du bouton "suivant" est déterminé par une propriété de APPLI.REF\publication.properties :
jway.xsl.param.actionButtonNextStep=**next**

Les valeurs acceptées sont :

- next : passer à la page suivante
- continue : aller à la première page en erreur ou non remplie

Dans les publications **statiques**, les boutons de navigation (avant/arrière) sont redéfinis dans APPLI.REF\css\theme.css :

- 👉 ○ #mainzone div.step input.**previous**PageButton { background-color: purple; }
- #mainzone div.step input.**next**PageButton { background-color: pink; }

Les boutons désactivés portent la classe "disabled" :

- 👉 #mainzone div.step input.**disabled** { display: none; }

Les textes de ces boutons sont définis par :

- labelPreviousPage : Page précédente
- labelNextPage : Page suivante

(à ne pas confondre avec les labels Step des étapes des formulaires)

14. Notice d'aide

La notice du document affiche une popup modale en vue "**moreinfo**". La **balise** <body> de cette vue porte la classe "moreInfo". Ainsi, par exemple, dans APPLI.REF\css\theme.css :

- 👉 body.**moreInfo** div#layout_center {
background-color: lightblue;
}

Bien sûr, d'autres attributs CSS peuvent être modifiés, comme la police de caractères.

Le **titre** du document :

- 👉 body.**moreInfo** div.documentTitles div.**chapter**Title {
background-color: blue;
color: white;
}

Là aussi, le champ des variations CSS est grand ouvert.

La **section** en cours :

- 👉 body.**moreInfo** div.documentTitles div.**Title** {
color: red;
text-align: center;
}

Dans la **Notice** d'aide, les boutons de navigation sont redéfinis dans APPLI.REF\css\theme.css :

- 👉 ○ body.moreInfo div.navigation span.previousPage a.previousPageLink { font-family: cursive; background-color: yellow; }
- body.moreInfo div.navigation span.previousPage**Disabled** { text-decoration: line-through; }
- body.moreInfo div.navigation span.nextPage a.nextPageLink { font-family: cursive; background-color: green; }
- body.moreInfo div.navigation span.nextPage**Disabled** { text-decoration: line-through; display: inline; }

Leurs labels sont :

- labelPreviousPage : Page précédente
- labelNextPage : Page suivante

(à ne pas confondre avec les labels Step des étapes des [formulaire](#)s)

15. Glossaire

Le glossaire apparait en pop up. Les définitions sont placées dans un Content avec la classe CSS "Content-definition", redéfinissable dans APPLI.REF\css\theme.css.

16. Action sur validation

Exemple de paramétrage pour :

- nom de datastore "interview.xml"
- dépôt sur validation dans le dossier /myfolder du serveur
- datastore de type "strict"

Ajouter les lignes suivantes à APPLI.REF\publication.properties :

```
jway.datastore.format=xml
jway.datastore.file.name=interview
jway.datastore.file.extension=xml
jway.MixedPortal.valid.mode=store
jway.MixedPortal.pathstore=/myfolder
jway.MixedPortal.valid.join.ds=strict
```

Bien sûr le processus serveur doit avoir le droit d'écrire dans le dossier jway.MixedPortal.pathstore

17. Conclusion

En suivant pas à pas ce tutorial-cookbook, un "auteur" ayant suivi la formation initiale FormPublisher peut ajuster l'ergonomie et le rendu graphique des pages [HTML](#) des applications web qu'il conçoit et réalise sans programmation.
