

**Custom Coach View Guide BPM**

**21 mai 2015**

**TABLE DES MATIÈRES**

[Contrôle(coach view) maison 3](#_Toc421173130)

[Text (blc) 3](#_Toc421173131)

[Text Area(blc) 3](#_Toc421173132)

[Select (blc) 3](#_Toc421173133)

[Date Time Picker (blc) 4](#_Toc421173134)

[ImageForRowSelect (blc) 4](#_Toc421173135)

[ImageForDocumentSelect (blc) 4](#_Toc421173136)

[AgeCalculator (blc) 5](#_Toc421173137)

[Boundary Trigger Event (blc) 5](#_Toc421173138)

[ButtonClickListener (blc) 5](#_Toc421173139)

[Confirmation Dialog (blc) 5](#_Toc421173140)

[Error Message Display Dialog (blc) 5](#_Toc421173141)

[Information Message Display Dialog (blc) 6](#_Toc421173142)

[Confirmation Html Dialog (blc) 6](#_Toc421173143)

[Dynamic HTML (blc) 6](#_Toc421173144)

[ExportCSV (blc) 6](#_Toc421173145)

[Field Set (blc) 6](#_Toc421173146)

[File Uploader (blc) 6](#_Toc421173147)

[Padding Formatter (blc) 7](#_Toc421173148)

[Padding Formatter (blc) 7](#_Toc421173149)

[Dual List (blc) 7](#_Toc421173150)

[List (blc) 8](#_Toc421173151)

# Contrôle(coach view) maison

Voici la liste des « coachview » qui ont été modifiés ou créés pour la BLC. Pour répondre à certains besoin particulier, plusieurs contrôles sont ceux du BPM avec quelques modifications.

## Text (blc)

Le contrôle de texte représente le « textbox » classique. Nous avons ajouté certaines fonctionnalités à celui-ci. Voici la liste :

* La visibilité « Required » a été modifié afin que celle-ci ne fasse qu’afficher un astérix (\*) devant le coach view. Cet ajustement était nécessaire car il causait un problème lorsque nous avions un validation au niveau des champs et business dans le même coach. La fonctionnalité enlevé n’empêchait pas l’utilisateur de cliquer sur un bouton. N’étant qu’une nuisance, elle a été enlevé.
* Le FIC ne supportant que les caractères en majuscule sans accents, nous avons donc ajouté un convertion du texte lorsque le « focus » sort du « coachview ». Toutes les lettres minuscule sont donc converti en majuscule et les lettres avec accents converties en majuscule sans accents.
* Nous avons ajouté la configuration « maxLength ». Celle-ci permet de limiter le nombre de caractères que l’usager peut saisir.
* Nous avons ajouté la propriété « textFormat ». Permet d’afficher le format de saisi attendu en dessous du « coach view ». Peut-être utile pour aider l’usager à saisir le bon format au lieu qu’il reçoive une erreur à la sauvegarde.

## Text Area(blc)

Le contrôle « Text Area » est un contrôle de saisi de longue chaîne de caractère. Nous avons ajouté certaines fonctionnalités à celui-ci. Voici la liste :

* La visibilité « Required » a été modifié afin que celle-ci ne fasse qu’afficher un astérix (\*) devant le coach view. Cet ajustement était nécessaire car il causait un problème lorsque nous avions un validation au niveau des champs et business dans le même coach. La fonctionnalité enlevé n’empêchait pas l’utilisateur de cliquer sur un bouton. N’étant qu’une nuisance, elle a été enlevé.
* Le FIC ne supportant que les caractères en majuscule sans accents, nous avons donc ajouté un convertion du texte lorsque le « focus » sort du « coachview ». Toutes les lettres minuscule sont donc converti en majuscule et les lettres avec accents converties en majuscule sans accents.
* Nous avons ajouté la configuration « maxLength ». Celle-ci permet de limiter le nombre de caractères que l’usager peut saisir.

## Select (blc)

Le contrôle « Select » représente la liste déroulante classique. Nous avons ajouté certaines fonctionnalités à celle-ci. Voici la liste :

* La visibilité « Required » a été modifié afin que celle-ci ne fasse qu’afficher un astérix (\*) devant le coach view. Cet ajustement était nécessaire car il causait un problème lorsque nous avions un validation au niveau des champs et business dans le même coach. La fonctionnalité enlevée n’empêchait pas l’utilisateur de cliquer sur un bouton. N’étant qu’une nuisance, elle a été enlevée.
* Contrairement au « Select » standard, il est possible de « bindé » une valeur « String » et le « Select » utilisera celle-ci pour sélectionner la bonne valeur dans la liste d’objet qui lui est liée. Ceci permet aussi de retourné que la valeur sélectionné plutôt que de retourner un objet complex. Par exemple, on peut lié une liste d’objet « NameValuePaired » et une valeur sélectionné « String ». Le « Select » d’origine oblige que l’objet sélectionné soit du même type que les objets contenu dans la liste lié (il est aussi possible de faire ça avec la version blc).
* Nous avons ajouté la configuration « noSelectionMessage ». Permet d’afficher un message lorsqu’aucune sélection n’a été faite au lieu de laisser le « Select » vide.

## Date Time Picker (blc)

Le contrôle « Date Time Picker » représente le contrôle de sélection de date. Nous avons ajouté certaines fonctionnalités à celle-ci. Voici la liste :

* La visibilité « Required » a été modifié afin que celle-ci ne fasse qu’afficher un astérix (\*) devant le coach view. Cet ajustement était nécessaire car il causait un problème lorsque nous avions un validation au niveau des champs et business dans le même coach. La fonctionnalité enlevée n’empêchait pas l’utilisateur de cliquer sur un bouton. N’étant qu’une nuisance, elle a été enlevée.
* Sur le contrôle original, dès que l’usager clique n’importe où sur le « coachview », le calendrier de sélection s’affiche. Nous avons modifié le comportement afin que le calendrier ne s’affiche que sur l’icône calendrier.
* Lorsque le « coachview » est saissable, nous avons modifié le format de saisi et ajouté un format sous le « coachview » indiquant à l’utilisateur comment la date doit être saisi. Le format de saisi valide est différent en français et en anglais.
* Lorsque le « coachview » est en lecture seul, nous avon modifié le format de saisi et ajouté un format sous le « coachview » indiquant à l’utilisateur comment la date doit être saisi. Le format de saisi valide est différent en français et en anglais. Il est à noter que ce format est différent que celui de la saisi dans les 2 langues.

## ImageForRowSelect (blc)

Le contrôle « ImageForRowSelect » développé maison permet sélectionner la ligne d’une grille et sortir du « coach ». Sont apparence est une image sélectionnable par le développeur. Il ne peut être utilisé que dans une grille (table). Il est principalement utilisé comme un bouton permettant de modifier, supprimer ou autre action sur la ligne de la grille où il se trouve.

* La configuration « imageName**»** permet de spécifier le fichier image à utiliser. Si aucune image n’est spécifié, le contrôle ne sera pas visible
* La configuration « visibility » permet de spécifier si le bouton est visible ou pas. Pour des raisons dont je ne me souviens pas, la valeur « INVISIBLE » cache le bouton et n’importe qu’elle autre valeur l’affiche. Il est à noter que pour fonctionner, il faut aussi assigner la variable de visibilité à la liaison « binding » du contrôle.

## ImageForDocumentSelect (blc)

Le contrôle « ImageForDocumentSelect » développé maison permet d’afficher un documentsélectionner la ligne d’une grille et sortir du « coach ». Sont apparence est une image sélectionnable par le développeur. Il ne peut être utilisé que dans une grille (table). Il est principalement utilisé comme un bouton permettant de modifier, supprimer ou autre action sur la ligne de la grille où il se trouve.

* La configuration « imageName**»** permet de spécifier le fichier image à utiliser. Si aucune image n’est spécifié, le contrôle ne sera pas visible.
* La configuration « objectId » permet de spécifier le document à afficher.
* La configuration « blcDocumentId » ne semble pas servir. Ne rien mettre
* La configuration « retrieveDocumentService » est le service ajax à utiliser pour trouver le document.
* La configuration « noDocErrorMsg » permet de spécifier le message qui sera afficher si le document n’a pas été trouvé.

## AgeCalculator (blc)

Le contrôle « AgeCalculator » développé maison permet de calculer l’âge à partir d’une date.

* La configuration « fromValue**»** étant de format « Date » permet de spécifier la date de naissance à utiliser.
* La configuration « toValue » étant de format « String » permet de spécifier la variable dans laquelle l’âge sera retournée.

## Boundary Trigger Event (blc)

Le contrôle « Boundary Trigger Event » a été importé d’un tierce toolkit. Il permet de surveiller une valeur et sortir du coach lorsque celle-ci change. Il est présentement utilisé dans presque tous les écrans pour gérer la dynamique.

* La configuration « fromValue**»** étant de format « Date » permet de spécifier la date de naissance à utiliser.
* La configuration « toValue » étant de format « String » permet de spécifier la variable dans laquelle l’âge sera retournée.

## ButtonClickListener (blc)

Le contrôle « ButtonClickListener » est développé maison. Il peut être très difficile de gérer des validations différentes dans le même écran. Si plus d’un bouton génère une validation dans un même écran, il faut être en mesure de savoir lequel a été appuyé. Pour utiliser ce contrôle, il suffit de créer une valeur booléenne pour chaque bouton et les « binder » au « ButtonClickListenet » ainsi qu’à chaque bouton. Dans la boîte de validation, la valeur booléenne à « true » est celle du bouton qui a été appuyé. Pour le moment, seulement 5 boutons sont supportés mais il est possible de modifier le « coach view » pour en ajouter.

* La configuration « button1Click**»** étant de format « boolean » contient la « flag » du premier bouton.
* La configuration « button2Click**»** étant de format « boolean » contient la « flag » du deuxième bouton.
* …

## Confirmation Dialog (blc)

Le contrôle « Confirmation Dialog » est développé maison à partir de « coach view » importé d’un tierce toolkit. Il permet d’afficher des messages de confirmation dans un petit dialogue par-dessus l’écran courant.

* La configuration « errorMessage**»** représente le message de confirmation à afficher.
* La configuration « showBinding**»** étant de format « boolean » contient la « flag » indiquant si le dialogue est affiché ou non.

## Error Message Display Dialog (blc)

Le contrôle « Error Message Display Dialog » est développé maison à partir de « coach view » importé d’un tierce toolkit. Il permet d’afficher des messages d’erreur dans un petit dialogue par-dessus l’écran courant.

* La configuration « errorMessage**»** représente le message d’erreur à afficher.
* La configuration « showBinding**»** étant de format « boolean » contient la « flag » indiquant si le dialogue est affiché ou non.

## Information Message Display Dialog (blc)

Le contrôle « Information Message Display Dialog » est développé maison à partir de « coach view » importé d’un tierce toolkit. Il permet d’afficher des messages d’information dans un petit dialogue par-dessus l’écran courant.

* La configuration « errorMessage**»** représente le message d’information à afficher.
* La configuration « showBinding**»** étant de format « boolean » contient la « flag » indiquant si le dialogue est affiché ou non.

## Confirmation Html Dialog (blc)

Le contrôle « Confirmation Html Dialog » est développé maison à partir de « coach view » importé d’un tierce toolkit. Il est identique au « Confirmation Dialog » à l’exception que le message de confirmation est de type HTML au lieu d’un simple « text » standard.

## Dynamic HTML (blc)

Le contrôle « Dynamic HTML » a été importé d’un tierce toolkit. Il permet d’afficher du code html et se rafraîchir automatiquement lorsque la valeur « bindé » change. Aucune variable de configuration n’existe pour ce contrôle.

## ExportCSV (blc)

Le contrôle « ExportCSV » est développé maison. Il permet d’exporter des données vers Excel dans un format (csv). La valeur « bindé » est une longue chaîne de caractères de type CSV qui sera envoyé à Excel.

* La configuration « cvsFileName**»** permet de spécifier le nom du document qui sera créé.
* La configuration « linkTitle**»** Permet de spécifier le libellé affiché à l’écran. Si aucune valeur n’est spécifié, « Export » sera affiché.

## Field Set (blc)

Le contrôle « Field Set » a été importé d’un tierce toolkit. Il s’agit d’un conteneur pour d’autres « coach view ». Il a un Il permet d’exporter des données vers Excel dans un format (csv). La valeur « bindé » est une longue chaîne de caractères de type CSV qui sera envoyé à Excel.

* La configuration « cvsFileName**»** permet de spécifier le nom du document qui sera créé.
* La configuration « linkTitle**»** Permet de spécifier le libellé affiché à l’écran. Si aucune valeur n’est spécifié, « Export » sera affiché.

## File Uploader (blc)

Le contrôle « File Uploader » a été importé d’un tierce toolkit et modifié pour répondre au besoin de la BLC. Il permet de mettre en voûte 1 document. Il est à noter que toutes les configurations ici-bas, doivent être indexé avant l’ouverture du coach.

* La configuration « legendTitle**»** permet de spécifier le libellé du qui sert de titre au « File Uploader »
* La configuration « submitLabel**»** permet de spécifier le libellé du bouton de téléchargement.
* La configuration « selectLabel » permet de spécifier le libellé du bouton permettant de parcourrir les répertoires pour choisir le document.
* La configuration « fileNameLabel » permet de spécifier le libellé de la colonne « fichier » de la grille du contrôle.
* La configuration « typeLabel » permet de spécifier le libellé de la colonne « type » de la grille du contrôle.
* La configuration « parameters » permet de spécifier les paramètres qui seront envoyé à la voûte pour créer le document. Il y a quelques paramètres qui sont obligatoire… les voicis :
* ecmServerConfigurationName (VOUTE\_ACCEO\_TK\_BLC)
* cmis:name
* objectTypeId (fcr:document)
* folderPath (/)
* snapshotId (tw.system.model.processApp.currentSnapshot.id)

## Padding Formatter (blc)

Le contrôle « Padding Formatter » a été est développé maison. Il permet d’ajouter un caractère X nombre de fois au début ou à la fin du chaîne de caractère. Le nombre de caractères ajoutés est la configuration « valueLength » de laquelle on soustrait la longueur de la longueur de la chaîne d’origine.

* La configuration « fromValue**»** représente la chaîne d’origine à modifier
* La configuration « toValue**»** représente le résultat de la modification. Ils est à noter que la même variable que la configuration « fromValue » est habituellement utilisée.
* La configuration « paddingCharacter » permet de spécifier le caractère à ajouté à la châine.
* La configuration « leftPadding » permet de spécifier si on ajoute des caractères au début ou à la fin.

## Padding Formatter (blc)

Le contrôle « Padding Formatter » a été est développé maison. Il permet d’ajouter un caractère X nombre de fois au début ou à la fin du chaîne de caractère. Le nombre de caractères ajoutés est la configuration « valueLength » de laquelle on soustrait la longueur de la longueur de la chaîne d’origine.

* La configuration « fromValue**»** représente la chaîne d’origine à modifier
* La configuration « toValue**»** représente le résultat de la modification. Ils est à noter que la même variable que la configuration « fromValue » est habituellement utilisée.
* La configuration « paddingCharacter » permet de spécifier le caractère à ajouté à la chaîne.

## Dual List (blc)

Le contrôle « Dual List » a été importé d’un tierce toolkit. Il s’agit d’un contrôle comprenant 2 listes dont une représente les items disponible et l’autre ceux sélectionnés.

* La configuration « sourceList**»** représente la liste des items disponibles
* La configuration « sourceList**»** représente la liste des items sélectionnés
* La configuration « sourceLabel**»** représente le libellé de la liste des items disponibles.
* La configuration « targetLabel**»** représente le libellé de la liste des items sélectionnés.
* La configuration « width**»** représente la largeur du coach view. Ne pas oublier de mettre « px » après la taille. Exemple : 500px
* La configuration « size**»** représente la hauteur du coach view.
* La configuration « showUpDownButtons » permet de spécifier si on veut pouvoir réordonner le contenu de la liste des items sélectionnés.

## List (blc)

Le contrôle « List » a été importé d’un tierce toolkit. Il s’agit d’un contrôle permettant de sélectionner un item.

* La configuration « sourceList**»** représente la liste des items disponibles
* La configuration « width**»** représente la largeur du coach view. Ne pas oublier de mettre « px » après la taille. Exemple : 500px
* La configuration « size**»** représente la hauteur du coach view.
* La configuration « multiple » permet de spécifier si on veut pouvoir sélectionner plus d’un item à la fois.