

Formulaires

Présentation

Dans un processus, Workey permet, à l'aide du concept d'état, de modéliser le cycle de vie d'un document (Gestion de contrats fournisseurs, Demande d'achat, Fiche produit, Réclamation client, ...) associé au document.

Dans un objectif d'automatisation et de dématérialisation d'un processus, il est nécessaire de dématérialiser aussi le document sous la forme d'un formulaire électronique.

Workey ne se limite pas à préciser les différents champs composant le document, mais il permet aussi de préciser les règles de visibilité (lecture, caché, édition) de l'information en fonction de différents critères : rôle, opération ou état du document.

Un document est composé d'un ensemble d'objets (Composants) contenant des champs qui sont rendus accessibles à l'utilisateur par l'intermédiaire de formulaires. Ainsi, il est possible d'associer à un document donné, plusieurs formulaires qui partageront cet ensemble de champs afin de gérer très finement les règles de visibilité en fonction du rôle, de l'opération réalisée et de l'état du document. Workey propose un outil de création de formulaire permettant :

- la création de **sections** (regroupement logique de champs)
- la création de **composants**
- la création de **tables** et de **tableaux dynamiques**

Ces composants possèdent une série de propriétés qui permettent de gérer leur comportement et par conséquent le comportement des formulaires.

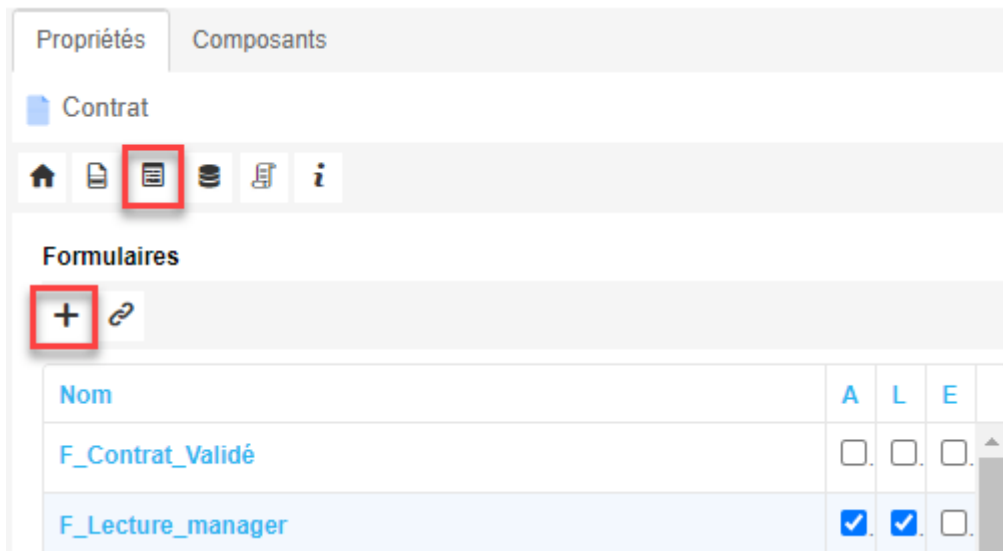
Après avoir modélisé le flux et le(s) formulaire(s), le modélisateur, par action sur n'importe quel document, peut choisir son formulaire. Dans le cas où il ne le ferait pas, Workey prendrait alors le premier formulaire créé et le positionnerait par défaut.

Création d'un formulaire

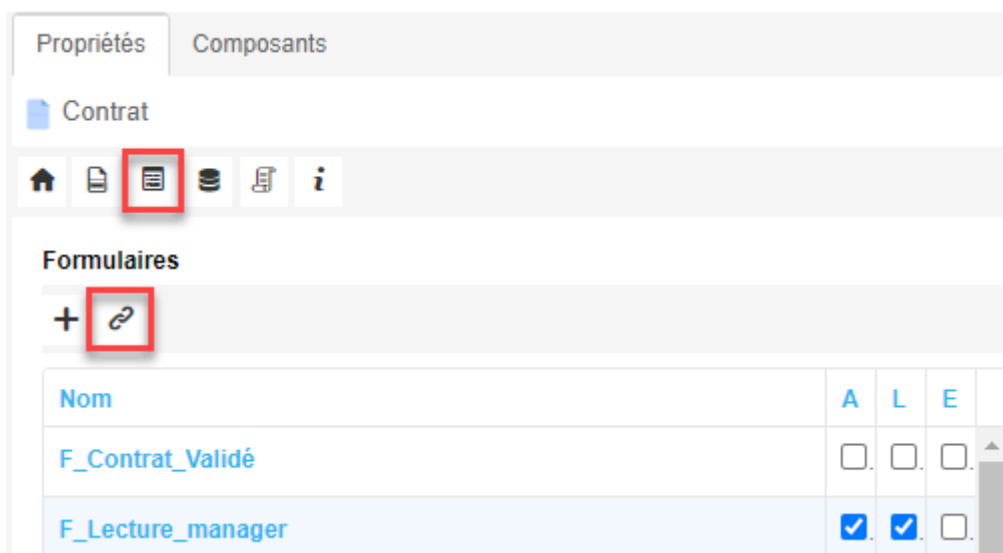
La création d'un formulaire associé à un document s'effectue soit depuis les propriétés d'un document, soit dans la section **Formulaires** du panneau de Projets.

A partir des propriétés d'un document

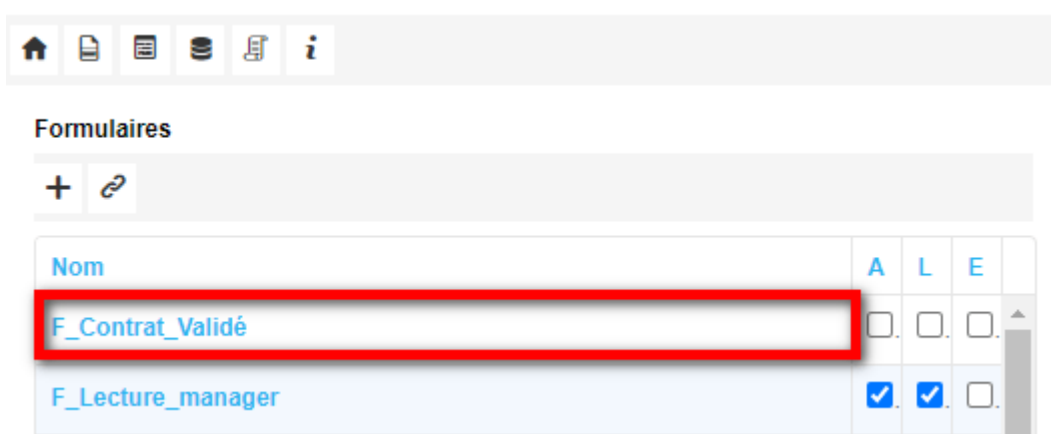
Pour créer un formulaire associé à un document dans un graphe de procédures ou processus il suffit de sélectionner les propriétés d'un document et cliquer sur le bouton **Formulaire** et ensuite sur le bouton **Nouveau** .



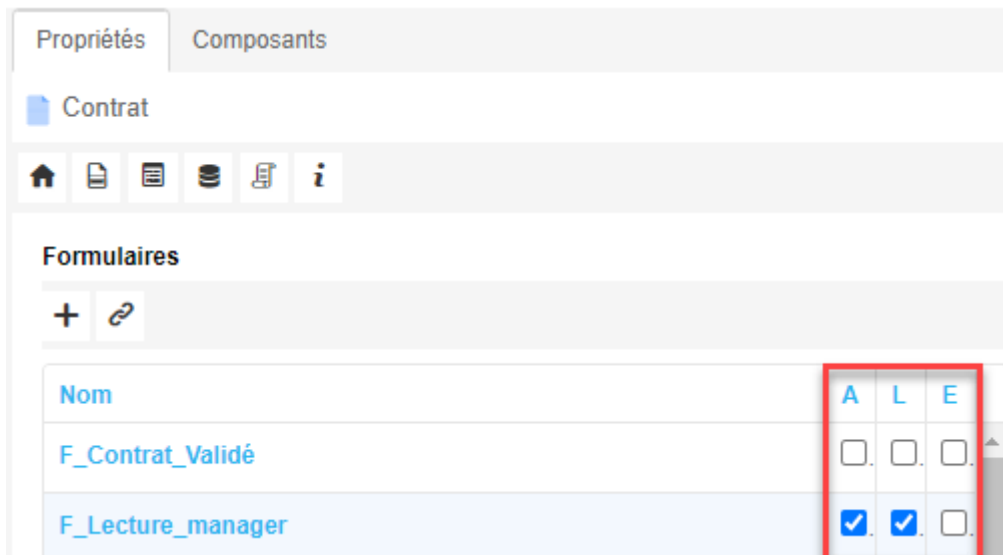
Il est possible également de **rattacher** un formulaire “orphelin” au document courant en cliquant sur le bouton Rattacher un formulaire inutilisé.



Pour **accéder** au formulaire existant ou en création il suffit de cliquer sur le lien existant dans la liste ce qui permettra d’ouvrir l’interface de conception du formulaire.



Les droits pour accéder au formulaire peuvent être définis également dans ce panneau:

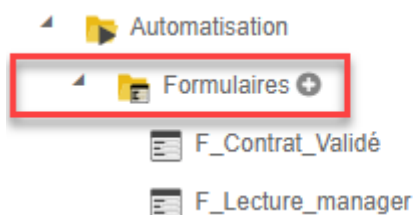


Ces droits sont identifiés avec une lettre dans l'en-tête de la colonne et correspondent aux définitions suivantes:

- **Archivage (A)** : Permet de définir le formulaire sélectionné comme celui qui sera présenté par le module d'archivage
- **Lecture par défaut (L)** : Permet de définir le formulaire sélectionné comme celui qui sera présenté par défaut en lecture
- **Écriture par défaut (E)** : Permet de définir le formulaire sélectionné comme celui qui sera présenté par défaut en écriture

A partir du panneau Projets

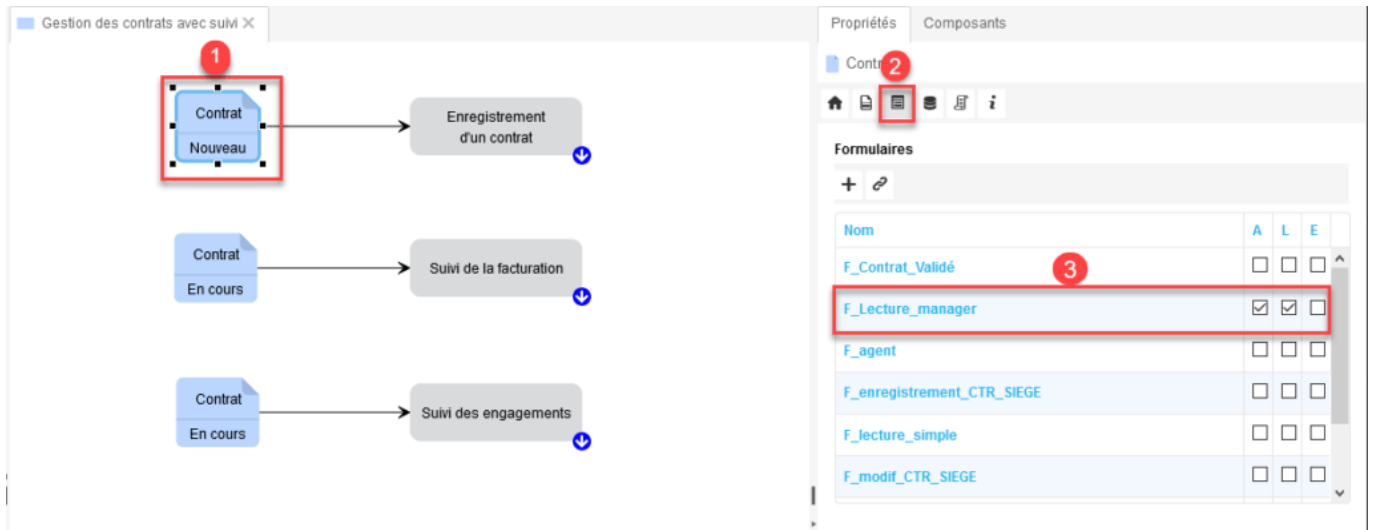
Il est possible également de cliquer sur la croix qui s'affiche à droite de la section **Formulaires** dans le panneau **Projets** pour créer un nouveau formulaire, puis l'associer au document souhaité.



Sélection d'un formulaire existant en entrée d'une opération

Pour sélectionner à une étape de la modélisation un formulaire spécifique, il suffit d'éditer les propriétés du lien qui devra porter ce formulaire en entrée d'une opération.

Sélectionner ensuite le formulaire.



Conseil

Pour afficher l'ensemble des noms de formulaire sur chaque lien d'une modèle de procédure, appuyer sur le touche "F". Pour masquer, appuyer à nouveau sur la touche "F"

Détermination du formulaire d'ouverture d'un document

Principes généraux

Définition

Le formulaire d'ouverture est le formulaire sélectionné par Workey lors de l'ouverture d'un document par un utilisateur. La sélection s'effectue en suivant les règles énoncées par le concepteur du processus.

Accès en lecture ou en écriture

En réponse à une demande d'ouverture d'un document, Workey est en mesure de déterminer si l'utilisateur peut accéder au document pour le modifier et changer son état, ou bien seulement pour une lecture de son contenu.

Détermination du rôle

La sélection du formulaire adéquat sera automatique dès que le rôle que peut et désire jouer l'utilisateur est connu. Dans la plupart des situations, un seul rôle est possible, ou bien plusieurs mais utilisant le même formulaire. Dans ce cas, l'ouverture du document s'effectue automatiquement avec le formulaire adapté, sans intervention de l'utilisateur.

Si un utilisateur joue plusieurs rôles opérationnels sur le document il se verra demander de choisir le rôle avant l'ouverture du document. Une fois celui-ci choisi, la détermination du formulaire est automatique.

Dans certains cas (l'utilisateur étant un "manager" du processus, ou bien

jouant plusieurs rôles opérationnels sur le document dans son état présent), l'utilisateur se verra demander de choisir le rôle pour l'ouverture du document. Un fois celui-ci choisi, la détermination du formulaire est automatique.

Utilisateurs, formulaire et vues

L'utilisateur peut également jouer un rôle de "workflow manager" ou de "process manager" pour le processus auquel participe le document. Dans ce cas, ce rôle est également proposé à l'utilisateur comme rôle d'ouverture, en plus des rôles opérationnels déterminés ci-dessus

Si l'utilisateur n'a pas un droit d'écriture sur le document, la liste des rôles lui permettant de lire le document est constituée.

S'il n'a qu'un seul rôle ou que ces rôles utilisent le même formulaire de lecture, le rôle d'ouverture est déterminé.

Sinon, l'utilisateur devra sélectionner parmi les rôles possibles. Il est également possible que l'utilisateur possède un droit de lecture "privé" sur le document, c'est à dire attaché à son identité d'acteur. Dans ce cas, le rôle et le formulaire d'ouverture seront automatiquement déterminés par la suite.

Si le document est ouvert en tant que workflow ou process manager, le formulaire d'ouverture est le formulaire spécifié comme le formulaire par défaut en écriture pour le type de document concerné.

Si le document est ouvert en tant qu'acteur d'un rôle opérationnel, le formulaire d'ouverture est celui éventuellement spécifié pour les opérations concernées par ce rôle.

Si aucun formulaire n'a été spécifié pour ces opérations, le formulaire par défaut en écriture pour le type de document concerné est retenu.

Si le document est ouvert par un utilisateur possédant un droit de lecture sur le document provenant d'un rôle opérationnel, le formulaire d'ouverture est celui spécifié par défaut pour ce rôle.

La sélection du rôle d'ouverture

Accès en écriture

L'utilisateur peut figurer parmi les prochains intervenants sur le document en voie d'ouverture. Il peut donc accéder en écriture au document pour y effectuer une ou plusieurs opérations, au titre de un ou plusieurs rôles. Pour chacune de ces opérations, le concepteur du processus peut avoir prévu un formulaire particulier, ce formulaire sera alors inséré dans la liste des formulaires possibles.

Sinon, le formulaire spécifié comme le formulaire par défaut pour le rôle effectuant l'opération sera inséré dans cette liste.

S'il n'y a qu'un seul rôle, ou que tous ces rôles utilisent le même formulaire, le rôle opérationnel d'ouverture est déterminé. Sinon, le rôle devra être sélectionné par l'utilisateur. Le cas le plus courant est donc très simple: l'utilisateur ouvre le document avec le formulaire prévu pour l'opération qu'il doit effectuer.

Prise en compte des rôles de "manager"

L'utilisateur peut également jouer un rôle de "workflow manager" ou de "process manager" pour le processus auquel participe le document. Dans ce cas, ce rôle est également proposé à l'utilisateur comme rôle d'ouverture, en plus des rôles opérationnels déterminés ci-dessus.

Accès en lecture par le rôle

Si l'utilisateur n'a pas un droit d'écriture sur le document, la liste des rôles lui permettant de lire le document est constituée. S'il n'a qu'un seul rôle ou que ces rôles utilisent le même formulaire de lecture, le rôle d'ouverture est déterminé. Sinon, l'utilisateur devra sélectionner parmi les rôles possibles.

Accès en lecture personnalisé

Il est également possible que l'utilisateur possède un droit de lecture "privé" sur le document, c'est à dire attaché à son identité d'acteur. Dans ce cas, le rôle et le formulaire d'ouverture seront automatiquement déterminés par la suite

La détermination du formulaire

Accès en écriture

Si le document est ouvert en tant que workflow ou process manager, le formulaire d'ouverture est le formulaire spécifié comme le formulaire par défaut en écriture pour le type de document concerné. Si le document est ouvert en tant qu'acteur d'un rôle opérationnel, le formulaire d'ouverture est celui éventuellement spécifié pour les opérations concernées par ce rôle. Si aucun formulaire n'a été spécifié pour ces opérations, le formulaire par défaut en écriture pour le type de document concerné est retenu.

Accès en lecture par le rôle

Si le document est ouvert par un utilisateur possédant un droit de lecture sur le document provenant d'un rôle opérationnel, le formulaire d'ouverture est celui spécifié par défaut pour ce rôle.

Propriétés d'un formulaire

Les propriétés de chaque formulaire sont affichées lors de sa sélection dans l'interface.

Propriétés Composants

F_Lecture_manager

🏠 📄 ⓘ

Libellé

F_Lecture_manager

Nom interne

F_Lecture_manager

Nom int. calculé automatiquement

Document

Contrat

Aperçu des pièces jointes

Activer l'aperçu

Pièce jointe à prévisualiser par défaut

Général

Libellé : Permet de saisir un libellé pour le formulaire

Nom interne : Attribut design_name de l'objet

Nom calculé automatiquement : Coché par défaut, cette option permet de garantir l'unicité des noms d'objets. **Document** : Permet de sélectionner le document associé au formulaire. Par défaut c'est le document pour lequel est défini le formulaire

Description Permet de saisir une description détaillée pour l'opération ainsi qu'un texte d'aide. Étendue permet de pointer vers un fichier externe contenant le texte de description.

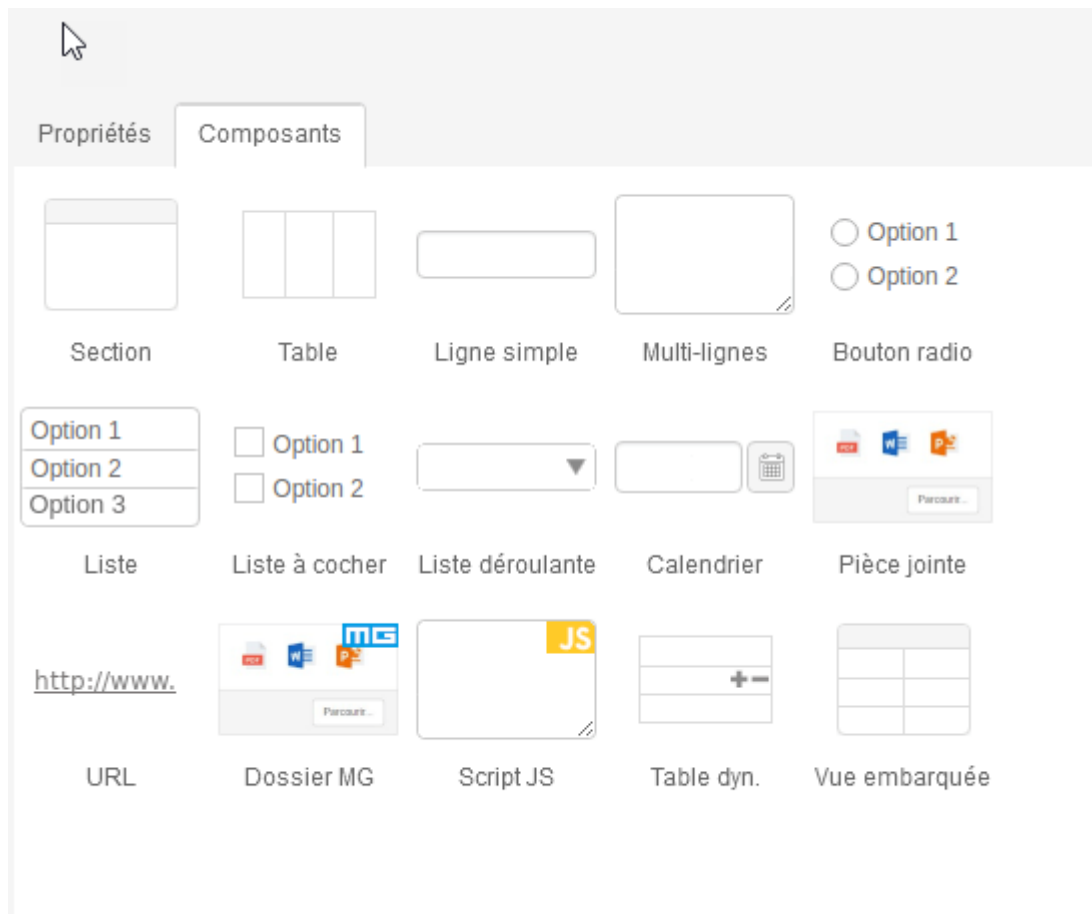
Liens

Il est aussi possible de joindre des liens logiques dans les propriétés des formulaires.

Composants d'un formulaire

Workey Designer propose plusieurs objets ou composants à placer et personnaliser pour composer vos formulaires.

Les objets qu'il est possible d'utiliser dans les formulaires sont affichés lorsque l'utilisateur clique sur l'onglet "Composants".



Pour placer, un composant afin de créer un formulaire, il vous suffit de glisser/déposer le composant.

Règles

- Aucun composant ne peut être placé à l'extérieur d'une section
- Certains types de champs sont interdits dans un tableau dynamique

Pour déplacer un composant au sein d'une section, inter section, table, table dynamique, il vous suffit de glisser/déplacer le composant.

Un indicateur visuel, horizontal ou vertical lors de la phase "glisser", vous permet de connaître l'emplacement futur du composant, une fois que vous avez "déposé" le composant.

Chaque composant sélectionné dispose de **propriétés** sont groupées dans six chapitres:



1. Général
2. Réglages
3. Scripts
4. Connecteurs

5. Description/Aide

6. Informations

Selon le type de composant plusieurs propriétés peuvent être proposées :

- **description** et texte d'aide
- **type** (texte, nombre, date, ...)
- **mono** ou **multivalué**
- **apparence** (zone de texte, radio bouton, ...)
- **domaine** de valeurs (liste de valeurs fixes ou dynamiques)
- **contrôle** (règle de gestion sur le champ)
- **valeur par défaut**
- règle de **visibilité** (lecture, caché, édition, obligatoire)
- gestion de l'**historique** sur les modifications des valeurs
- inclus ou non dans l'**index** (pour la recherche et les rapports)
- **synchronisé** ou non (comportement lors de synchronisations)
- **liens** (descriptions et présentations)
- **connecteurs** (Java, SQL, stockée)

Les paragraphes suivants décrivent les différents objets qui peuvent être utilisés dans les formulaires ainsi que ses caractéristiques. Cette description montre pour chaque objet les onglets caractéristiques à chaque objet et qui permettent de le configurer.

Sur chaque objet sélectionné les propriétés proposées peuvent varier, néanmoins il existe deux onglets communs à tous les objets : Description/Aide et Informations. Ces deux onglets sont décrits ci-après :

The image displays two side-by-side screenshots of a software configuration interface for a component named 'F_Contrat_Validé'. Both screenshots show the 'Propriétés' (Properties) tab. The left screenshot highlights the 'Home' icon (a house symbol) in the top navigation bar. The right screenshot highlights the 'Info' icon (an 'i' symbol) in the same bar. Below the navigation bar, the left screenshot shows fields for 'Libellé' (labeled 'F_Contrat_Validé'), 'Nom interne' (labeled 'F_Contrat_Validé'), a toggle for 'Nom int. calculé automatiquement' (checked), 'Document' (set to 'Contrat'), 'Aperçu des pièces jointes' (with an unchecked 'Activer l'aperçu' checkbox), and 'Pièce jointe à prévisualiser par défaut'. The right screenshot shows fields for 'Id (master)' (containing 'FRME516AE452260C4F1ABDE5C897F44BFDA'), 'Id (instance)', 'MetaType' (set to 'FORM'), and 'GadgetType'.

Section

Une section permet de regrouper un ensemble de champs au sein d'une zone qui peut être soit dépliée ou pliée. Aucun composant ne peut être placé à l'extérieur d'une section.

Une section comporte un libellé.

L'apparence permet de préciser :

- les règles de visibilité de la section, et donc des différents champs qui la compose
- son apparence lors de l'ouverture du formulaire

Propriétés

The image displays two side-by-side screenshots of the 'Propriétés' (Properties) dialog box for a 'Conditions générales' component. Both screenshots show the 'Propriétés' and 'Composants' tabs. The left screenshot shows the 'Libellé' (Label) field with the text 'Conditions générales' and the 'Nom interne' (Internal name) field with the text 'Conditions_generales'. Below these fields is a toggle switch for 'Nom int. calculé automatiquement' (Internal name calculated automatically), which is currently turned on. The right screenshot shows the 'Règles de visibilité' (Visibility rules) section. It features three radio buttons: 'Lecture/Ecriture' (selected), 'Lecture seule', and 'Caché'. Below these is a checked checkbox for 'Minimiser la section'.

Ligne simple

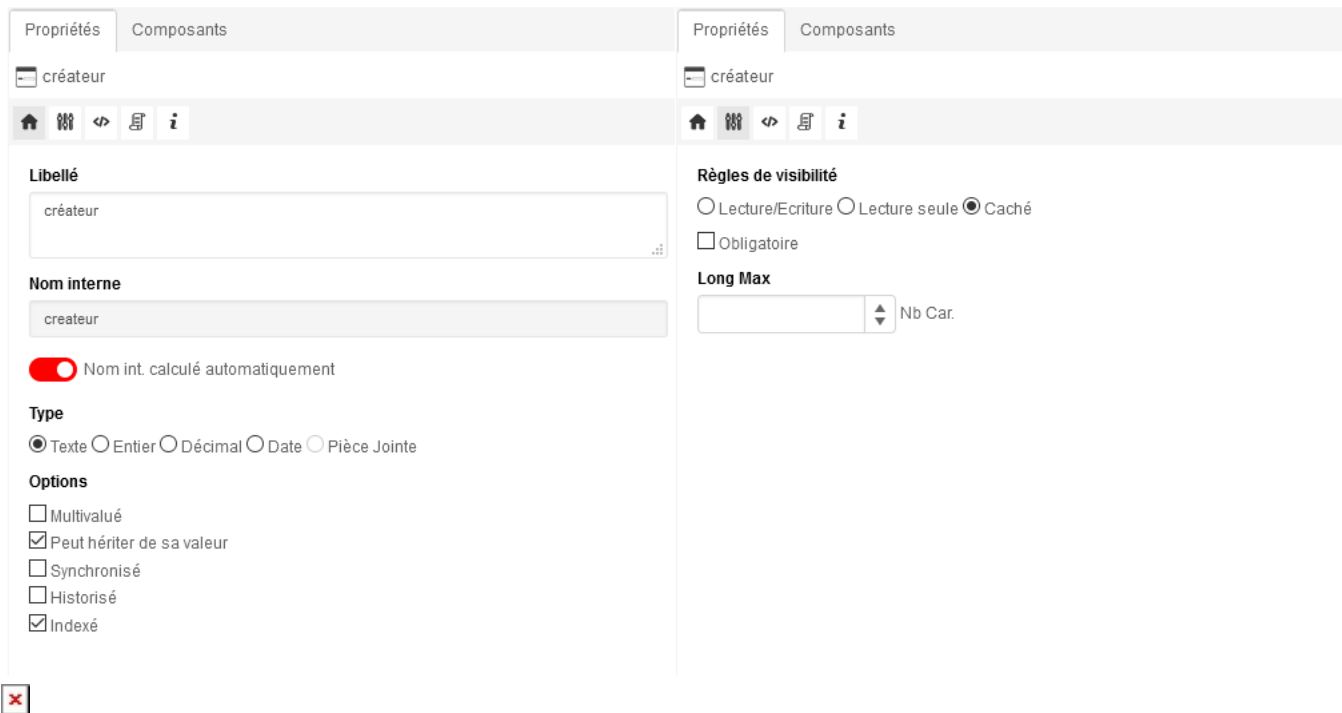
Le composant ligne simple est une zone de saisie où l'utilisateur peut entrer du texte. Ce champ peut aussi avoir une valeur par défaut ou un domaine de valeurs que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en lecture seule ou en écriture avec une taille personnalisée.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraichissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés



Multi-lignes

Le champ Multi-lignes est une zone de texte (textarea) où l'utilisateur peut entrer des paragraphes de texte. Son **type** est obligatoirement texte. Ce champ peut aussi avoir une valeur par défaut que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en lecture seule ou en écriture avec une taille personnalisée.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraichissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés



Propriétés

Composants

☰ Multi-lignes 2

🏠
🔍
<>
📄
i

Type d'initialisation

Valeur statique

Expression calculée une seule fois

Expression calculée à chaque ouverture

Valeur/Script d'initialisation ✎

Script de contrôle ✎

Bouton ratio

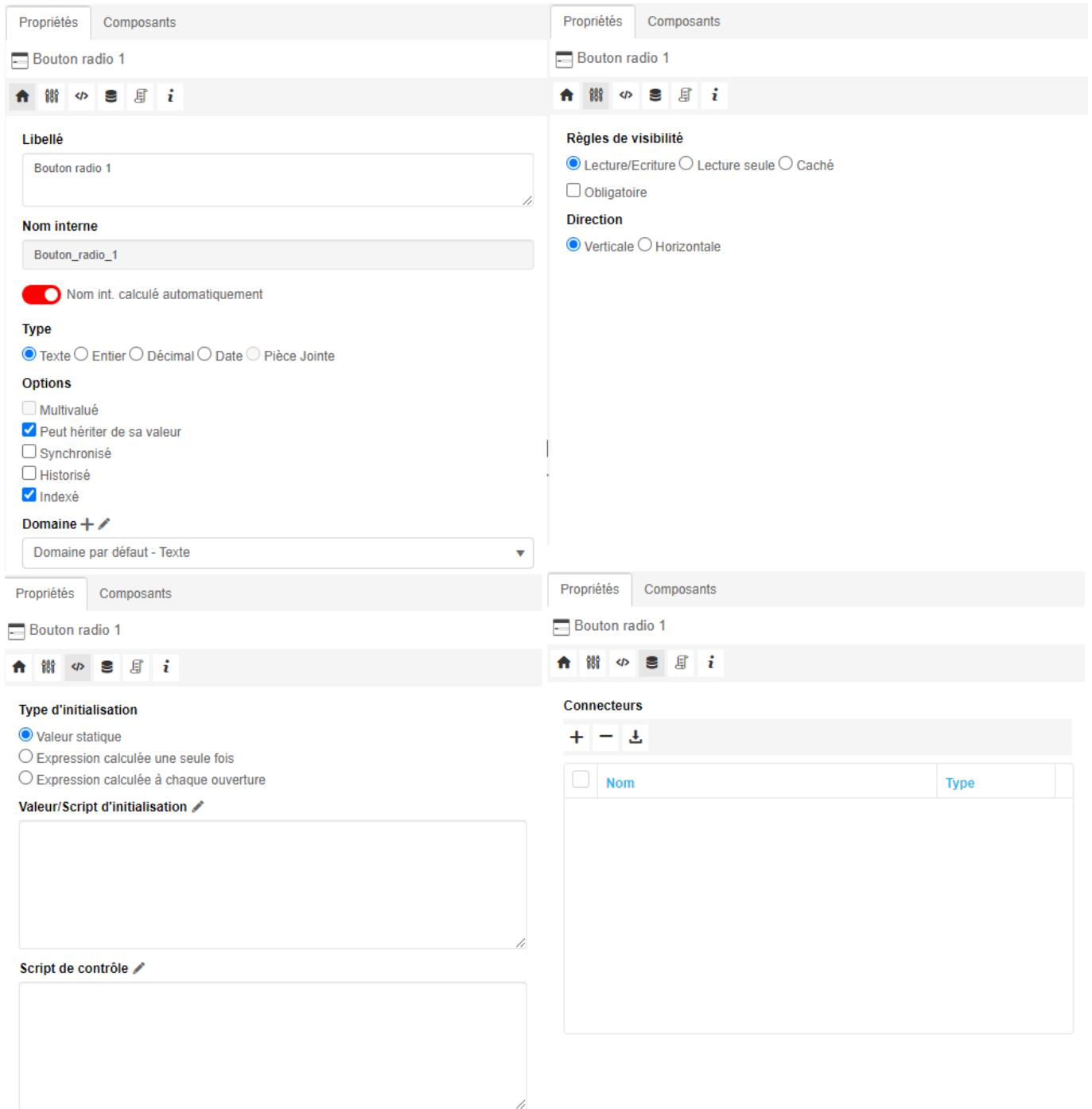
Le champ Bouton radio est une liste de choix dans laquelle l'utilisateur pourra sélectionner un élément. Son **type** est obligatoirement texte. Ce champ peut une valeur par défaut que l'utilisateur n'aura pas nécessairement besoin de modifier et doit obligatoirement avoir un domaine de valeurs associé. Il peut être présenté en lecture seule ou en écriture. On peut aussi le définir comme obligatoire, le formulaire ne serait pas soumis si sa valeur était vide.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraîchissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés



Liste

Le champ Liste est une liste de choix dans laquelle l'utilisateur pourra sélectionner un ou plusieurs éléments. Son **type** est obligatoirement texte. L'utilisateur peut effectuer une multi sélection des valeurs en maintenant la touche Ctrl du clavier appuyée.

Ce champ peut une valeur par défaut que l'utilisateur n'aura pas nécessairement besoin de modifier et doit obligatoirement avoir un domaine de valeurs associé. Il peut être présenté en lecture seule ou en écriture. On peut aussi le définir comme obligatoire, le formulaire ne serait pas soumis si sa valeur était vide.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à

chaque rafraîchissements et à la soumission de la page.
Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.
Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

The image displays four screenshots of the 'Propriétés' (Properties) window for a 'Liste 1' component, arranged in a 2x2 grid. Each screenshot shows a different tab of the configuration interface.

- Top-left screenshot:** Shows the 'Libellé' (Label) tab. The 'Libellé' field contains 'Liste 1'. The 'Nom interne' (Internal Name) field contains 'Liste_1'. A toggle switch for 'Nom int. calculé automatiquement' is turned on. The 'Type' (Type) is set to 'Texte' (Text). Under 'Options', 'Multivalué' (Multivalued), 'Peut hériter de sa valeur' (Can inherit its value), and 'Indexé' (Indexed) are checked. The 'Domaine' (Domain) is set to 'Domaine par défaut - Texte'.
- Top-right screenshot:** Shows the 'Règles de visibilité' (Visibility Rules) tab. The 'Lecture/Ecriture' (Read/Write) radio button is selected. Other options include 'Lecture seule' (Read-only), 'Caché' (Hidden), and 'Obligatoire' (Required), which is unchecked.
- Bottom-left screenshot:** Shows the 'Type d'initialisation' (Initialization Type) tab. The 'Valeur statique' (Static value) radio button is selected. Other options are 'Expression calculée une seule fois' (Expression calculated once) and 'Expression calculée à chaque ouverture' (Expression calculated on each opening). The 'Valeur/Script d'initialisation' (Value/Initialization Script) and 'Script de contrôle' (Control Script) fields are empty.
- Bottom-right screenshot:** Shows the 'Connecteurs' (Connectors) tab. It features a table with columns 'Nom' (Name) and 'Type' (Type). The table is currently empty.

Liste à cocher

Le champ Liste à cocher est une liste de choix dans laquelle l'utilisateur pourra choisir un ou plusieurs éléments. Son **type** est obligatoirement texte. Ce champ peut avoir une valeur par défaut vide ou prédéfinie que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être

présenté en lecture seule ou en écriture. On peut aussi le définir comme obligatoire, le formulaire ne serait pas soumis si sa valeur était vide. La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraîchissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

The image displays four screenshots of the 'Propriétés' (Properties) panel for a 'Liste à cocher 1' (Checkbox List 1) component. Each screenshot shows a different section of the configuration interface.

Screenshot 1 (Top Left): Shows basic properties including 'Libellé' (Label) set to 'Liste à cocher 1', 'Nom interne' (Internal Name) set to 'Liste_a_cocher_1', and a toggle for 'Nom int. calculé automatiquement' (Internal Name calculated automatically) which is turned on. Under 'Type', 'Texte' (Text) is selected. Under 'Options', 'Multivalué' (Multivalued), 'Peut hériter de sa valeur' (Can inherit its value), and 'Indexé' (Indexed) are checked.

Screenshot 2 (Top Right): Shows 'Règles de visibilité' (Visibility Rules) with 'Lecture/Ecriture' (Read/Write) selected. 'Obligatoire' (Mandatory) is unchecked. Under 'Direction', 'Verticale' (Vertical) is selected.

Screenshot 3 (Bottom Left): Shows 'Type d'initialisation' (Initialization Type) with 'Valeur statique' (Static Value) selected. Below is a text area for 'Valeur/Script d'initialisation' (Value/Initialization Script) and another for 'Script de contrôle' (Control Script).

Screenshot 4 (Bottom Right): Shows the 'Connecteurs' (Connectors) section with a table for defining connectors.

	Nom	Type
<input type="checkbox"/>		

Liste déroulante

Le champ Liste déroulante est une liste de choix dans laquelle l'utilisateur pourra sélectionner un élément. Son **type** est obligatoirement texte. Ce champ peut avoir une valeur par défaut vide ou prédéfinie que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en **lecture** seule ou en **écriture**. On peut aussi le définir comme **obligatoire**, le formulaire ne serait pas soumis si sa valeur était vide. Un **Domaine** de valeurs doit obligatoirement y être associé. La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraîchissements et à la soumission de la page. Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur. Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

The image shows two side-by-side screenshots of a configuration interface for a 'Liste déroulante 1' (dropdown list) field. The left panel, titled 'Propriétés', contains the following settings:

- Libellé:** Liste déroulante 1
- Nom interne:** Liste_deroulante_1
- Nom int. calculé automatiquement:** Enabled (red toggle)
- Type:** Texte (selected), Entier, Décimal, Date, Pièce Jointe
- Options:** Multivalué (unchecked), Peut hériter de sa valeur (checked), Synchronisé (unchecked), Historisé (unchecked), Indexé (checked)
- Domaine:** Domaine par défaut - Texte

The right panel, titled 'Règles de visibilité', contains the following settings:

- Règles de visibilité:** Lecture/Ecriture (selected), Lecture seule, Caché
- Obligatoire:** Unchecked
- Complétion automatique:** Unchecked

Calendrier

Calendrier est un champ de type Date dans lequel l'utilisateur peut sélectionner une date spécifique. Son **type** est obligatoirement date. Ce champ peut avoir une valeur par défaut vide ou prédéfinie que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en **lecture** seule ou en **écriture**. On peut aussi le définir comme **obligatoire**, le formulaire ne serait pas soumis si sa valeur était vide. La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à

chaque rafraîchissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

The image shows two side-by-side screenshots of the 'Propriétés' (Properties) window for a field named 'Calendrier 1'. The left screenshot displays the 'Propriétés' tab with the following settings: 'Libellé' is 'Calendrier 1'; 'Nom interne' is 'Calendrier_1'; 'Nom int. calculé automatiquement' is checked; 'Type' is 'Date'; 'Options' include 'Peut hériter de sa valeur' (checked), 'Indexé' (checked), and 'Historisé' (unchecked). The right screenshot shows the 'Règles de visibilité' (Visibility rules) section with 'Lecture/Ecriture' (Read/Write) selected, and 'Obligatoire' (Required) unchecked. The 'Masque de donnée' (Data mask) field is empty.

Pièce jointe

Le champ Fichier donne la possibilité à l'utilisateur d'attacher au formulaire un ou plusieurs fichiers externes.

Son **type** est obligatoirement pièce jointe. Ce champ peut avoir une valeur par défaut vide ou prédéfinie que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en **lecture** seule ou en **écriture**. On peut aussi le définir comme **obligatoire**, le formulaire ne serait pas soumis si sa valeur était vide.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraîchissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

The image shows a configuration window for a 'Pièce jointe 1' (Attachment 1) field. The window is divided into two main sections: 'Propriétés' (Properties) on the left and 'Composants' (Components) on the right.

Propriétés (Properties):

- Libellé (Label):** A text input field containing 'Pièce jointe 1'.
- Nom interne (Internal Name):** A text input field containing 'Piece_jointe_1'.
- Nom int. calculé automatiquement (Internal name calculated automatically):** A toggle switch that is currently turned off.
- Type (Type):** Radio buttons for 'Texte' (Text), 'Entier' (Integer), 'Décimal' (Decimal), 'Date', and 'Pièce Jointe' (Attachment). 'Pièce Jointe' is selected.
- Options (Options):**
 - Multivalué (Multivalued)
 - Peut hériter de sa valeur (Can inherit its value)
 - Synchronisé (Synchronized)
 - Historisé (Historical)
 - Indexé (Indexed)

Composants (Components):

- Règles de visibilité (Visibility Rules):**
 - Lecture/Ecriture (Read/Write)
 - Lecture seule (Read only)
 - Caché (Hidden)
 - Obligatoire (Required)
- Taille maximale par fichier (Maximum file size):** A numeric input field with a unit dropdown set to 'Mo' (Mo).
- Vignette (Thumbnail)

URL

Le champ URL permet d'insérer dans le formulaire un lien vers un site ou une page externe. Son **type** est obligatoirement texte.

Ce champ peut avoir une valeur par défaut vide ou prédéfinie que l'utilisateur n'aura pas nécessairement besoin de modifier. Il peut être présenté en **lecture** seule ou en **écriture**. On peut aussi le définir comme **obligatoire**, le formulaire ne serait pas soumis si sa valeur était vide.

La **Valeur** par défaut prise par ce champ au chargement de la page peut être vide ou prédéfinie. Une **Expression Stalk** sera évaluée une seule fois au chargement de la page. Une **Expression calculée** sera réévaluée au chargement, à chaque rafraîchissements et à la soumission de la page.

Une expression Stalk de **Contrôle** du champ peut assurer l'intégrité de la valeur.

Il est possible de lui associer un **Connecteur** Java, Sql ou une procédure stockée.

Propriétés

Propriétés Composants

URL 1

Libellé

URL 1

Nom interne

URL_1

Nom int. calculé automatiquement

Type

Texte Entier Décimal Date Pièce Jointe

Options

Multivalué

Peut hériter de sa valeur

Synchronisé

Historisé

Indexé

Propriétés Composants

URL 1

Règles de visibilité

Lecture/Ecriture Lecture seule Caché

Obligatoire

Table simple

Pour aligner des champs horizontalement vous pouvez utiliser le composant table. Une table peut intégrer tous les composants par glisser déposer à l'exception :

- d'une section
- d'une table
- d'un tableau dynamique
- d'une vue embarquée

La césure permet de spécifier le nombre de composants par ligne

Propriétés

Propriétés Composants

Table 7

Libellé

Table 7

Nom interne

Table_7

Nom int. calculé automatiquement

Nombre de colonnes

4 Colonne

Type

Texte Entier Décimal Date Pièce Jointe

Propriétés Composants

Table 7

Règles de visibilité sur les champs de la table

Lecture/Ecriture Lecture seule Caché

Tableau dynamique

Un **tableau dynamique** est constitué d'un ensemble de colonnes, correspondant chacune à un champ du document dans laquelle le tableau apparaît, et d'un ensemble de lignes correspondant aux différentes valeurs de ces champs. Pour le construire :

- glisser/déposer le composant tableau dynamique
- glisser/déposer un à un les composants que vous souhaitez intégrer dans le tableau afin de constituer les différentes colonnes du tableau

La cellule à l'intersection de la i-ème ligne et de la j-ème colonne représente la i-ème valeur du champ attribué à la j-ème colonne.

Une **table dynamique** est une table dans laquelle le nombre de lignes n'est pas pré-déterminé, mais varie suivant le choix de l'utilisateur qui peut ajouter ou enlever des lignes à la table en utilisant les boutons **ajouter** et **supprimer** à gauche du tableau.

Le champ Table dyn. permet de construire un tableau muni d'une infinité de lignes. Il peut être présenté en **lecture** seule ou en **écriture** ou **caché** à l'affichage.

On définit les champs composant les colonnes du tableau et c'est l'utilisateur qui créera autant de lignes qu'il lui est nécessaire.

Tous les champs présents à l'intérieur doivent être définis comme multivalués.

Les champs que l'on peut créer à l'intérieur d'un tel tableau sont : Ligne simple, Multi-ligne, Liste. Les valeurs par défaut ne sont pas prises en considération.

Ce type d'objet autorise aussi l'utilisation du langage STALK notamment pour effectuer des calculs sur une ligne ou bien par colonne.

Il est possible de pré-remplir des tableaux dynamiques par l'utilisation des pré-connecteurs.

Propriétés

Propriétés Composants

Table 21

🏠 📄 📄 ⓘ

Libellé

Table 21

Nom interne

Table_21

Nom int. calculé automatiquement

Type

Texte Entier Décimal Date Pièce Jointe

Propriétés Composants

Table 21

🏠 📄 📄 ⓘ

Règles de visibilité sur les champs de la table

Lecture/Ecriture Lecture seule

Vue embarquée

Le champ Vue embarquée permet de réserver dans un formulaire l'emplacement pour une vue construite dans une procédure. Il peut être présenté en **lecture** seule ou en **écriture** ou **caché** à l'affichage.

Les colonnes de la vue sont construites dans un graphe de procédures, pas dans un formulaire. On définit ici l'objet vue qui est associé au champ en le sélectionnant dans la liste déroulante des vues existantes. Il faut donc modéliser la vue avant de lui créer un champ de formulaire.

Propriétés

Propriétés Composants

Vue embarquée 1

🏠 📄 ⓘ

Libellé

Vue embarquée 1

Nom interne

Vue_embarquee_1

Nom int. calculé automatiquement

Type

Texte Entier Décimal Date Pièce Jointe

Vue embarquée

Contrôle d'intégrité des champs

Grâce au scripts STalk, vous pouvez entrer des règles de validation pour chaque champ. Ce contrôle est vérifié avant le changement d'état (Lors de

l'enregistrement). Notez que cette finalité est complémentaire aux règles de changement d'état. Toutes les valeurs et expressions sont des phrases STalk.

Intégration de code HTML et de JavaScript

Il est possible d'inclure du [code JavaScript](#) dans les formulaires. Celui-ci sera interprété par le navigateur lorsque l'objet le portant est chargé.

Ajoutez un champ multi-lignes dans le formulaire et éditez ses propriétés. Dans la section apparence choisir l'option "non éditable". Dans la section **Initialisation**, choisissez l'option **Valeur** et saisissez du code JavaScript ou HTML.

Pour une combinaison de champs internes au formulaire avec du code HTML, choisissez l'option Expression calculée pour que Workey évalue la valeur du ou des champs au rafraichissement de la page.

Exemple :

Visibilité des formulaires

Accès en lecture

L'utilisateur peut figurer parmi les prochains intervenants sur le document en voie d'ouverture. Il peut donc accéder en écriture au document pour y effectuer une ou plusieurs opérations, au titre de un ou plusieurs rôles. Pour chacune de ces opérations, le concepteur du processus peut avoir prévu un formulaire particulier, ce formulaire sera alors inséré dans la liste des formulaires possibles. Sinon, le formulaire spécifié comme le formulaire par défaut pour le rôle effectuant l'opération sera inséré dans cette liste. S'il n'y a qu'un seul rôle, ou que tous ces rôles utilisent le même formulaire, le rôle opérationnel d'ouverture est déterminé. Sinon, le rôle devra être sélectionné par l'utilisateur.

Le cas le plus courant est donc très simple: l'utilisateur ouvre le document avec le formulaire prévu pour l'opération qu'il doit effectuer

Accès en écriture

Si le document est ouvert en tant que workflow ou process manager, le formulaire d'ouverture est le formulaire spécifié comme le formulaire par défaut en écriture pour le type de document concerné. Si le document est ouvert en tant qu'acteur d'un rôle opérationnel, le formulaire d'ouverture est celui éventuellement spécifié pour les opérations concernées par ce rôle. Si aucun formulaire n'a été spécifié pour ces opérations, le formulaire par défaut en écriture pour le type de document concerné est retenu

Accès en lecture pour le rôle

Si le document est ouvert par un utilisateur possédant un droit de lecture sur le document provenant d'un rôle opérationnel, le formulaire d'ouverture est celui spécifié par défaut pour ce rôle.

Par rôle

Conseil

Pour définir un formulaire présenté par défaut à un rôle, éditer les propriétés de l'objet rôle dans une procédure. Dans la section Général, sélectionnez le formulaire désiré.

Prise en compte des rôles de manager

L'utilisateur peut jouer un rôle de "workflow manager" ou de "process manager" pour le processus auquel participe le document. Dans ce cas, ce rôle est également proposé à l'utilisateur comme rôle d'ouverture, en plus des rôles opérationnels déterminés ci-dessus.

Accès en lecture par le rôle

Si l'utilisateur n'a pas un droit d'écriture sur le document, la liste des rôles lui permettant de lire le document est constituée. S'il n'a qu'un seul rôle ou que ces rôles utilisent le même formulaire de lecture, le rôle d'ouverture est déterminé. Sinon, l'utilisateur devra sélectionner parmi les rôles possibles.

Accès en lecture personnalisée

Il est également possible que l'utilisateur possède un droit de lecture "privé" sur le document, c'est à dire attaché à son identité d'acteur. Dans ce cas, le rôle et le formulaire d'ouverture seront automatiquement déterminés par la suite.

Par opération

Pour définir un formulaire présenté par défaut pour une opération spécifique dans un conteneur de parallélisme, éditer les propriétés de l'objet **opération** dans conteneur de parallélisme. Dans la section Général, sélectionnez le formulaire désiré.

Par document

Pour associer un ou des formulaires à des objets document, ajoutez-les dans les propriétés du document. Dans la section Formulaires, choisissez le formulaire qui sera affiché par défaut :
en lecture en écriture