XML Pivot - Changelog implémentation "Connect"

Changelog du moteur XML Pivot implémentation "Connect"

1.59 - 18/04/2024

Nouveautés :

- **Source de données pour créer des paragraphes** :

  2 nouvelles sources de données (en plus d'un fichier texte) : contenu d'une liste ou d'une variable.

  Liste :

  `<paragraph list="list-name" ... [keep-blank-line="true"] ... >`

  Par défaut, c'est la présence de lignes blanches qui sert de séparateur de paragraphe (liste confectionnée à partir d'un texte déjà mis en forme, par exemple issu d'un spool texte paginé).
  Seul le premier champ de la liste est utilisé.
  Option `@keep-blank-line="true"` pour conserver les lignes blanches
  Exemple de liste pouvant être utilisée :

```
<list id="mymessage">
  <item type="msg"><field id="text">Dear customer,</field></item>
  <item type="msg"><field id="text"></field></item>
  <item type="msg"><field id="text">We hope this message finds you well. We would like to express our gratitude
  <item type="msg"><field id="text">continued support and loyalty to our brand.</field></item>
  <item type="msg"><field id="text"></field></item>
  <item type="msg"><field id="text">As part of our ongoing commitment to providing you with the best possible…</
  <item type="msg"><field id="text"></field></item>
  <item type="msg"><field id="text">John Smith</field></item>
</list>
```

  Variable :

  `<paragraph var="var-name" ... >`

  Dans tous les cas, le texte peut contenir des sauts de lignes forcés avec [[BR]] et tous les éléments dynamiques déjà pris en charge avec les sources de type fichier comme {{line:*xxxx*}} , {{field:*xxxx*}} , {{col:*xxxx*}} , [[*xxxx*]]

- **@item-number :**
  Prise en charge de la valeur "first" (équivalent à 1)
  Utilisation possible de la valeur « last » sur les éléments <line> en plus des éléments <field>

- **<cond> :**

  Plus besoin d'espaces autour de « = » et « != »
  Exemple :

  `<cond name="mycond">field(NUMSOC)="1234"</cond>`

  `<cond name="mycond">field(NUMSOC)!="1234"</cond>`

Possibilité de déclarer une condition toujours vraie ou fausse de manière intuitive et sans nécessité de doubles quotes autour des valeurs true et false :

Exemples :

```
<cond name="condname">true</cond>
<cond name="condname">false</cond>
```

- **<field trim="left" > :**

  Option de trim sur les field (trim="left" , trim="right" ou trim="full")

  Par défaut : pas de trim

- **@aggregate-function (ou @grouping-function) :**

  2 nouvelles fonctions sont disponibles sur les agrégats de listes :

  count() : nombre total d'item

  avg() : moyenne d'une valeur numérique

  Exemples :

```
<list-header>
  <field name="quantity" aggregate-function="count()" />
  <field id="amount" aggregate-function="avg()"/>
</list-header>
```

- **<list page-footer-reserve="true" > :**

  Option **@page-footer-reserve="true"**(sur liste de niveau 1) afin d'ajouter automatiquement une ligne fictive en fin de tableau permettant de réserver l'espace nécessaire à un pied de document fixe. En cas d'utilisation conjointe de l'option @indivisible, cette dernière <line> sera intégrée au dernier bloc (<group>) afin d'éviter les pieds de page orphelins.

Correctifs :

- Corrections de mise en forme des valeurs négatives
- Suppression de champs supplémentaires hors <line> en cas d'usage de list-header, list-footer ou list-aggregate
- Si élément <list> sans contenu explicite (<list id="xxx" />), nommage des <line> avec le type d'item dans le cas où cette information (@type) est présente dans le fichier de données au niveau des item.
- Dans le cadre d'un agrégat de liste, les<line> ne sont plus nommées avec le type d'item éventuel, dans le sens où les données proviennent de plusieurs item potentiellement de types différents.

Nouveautés techniques :

- Technique de mise en forme des numériques sans passer par des flottants afin de limiter au maximum les problèmes d'arrondis.

1.58 - 25/03/2024

Nouveautés :

- Aggregat de liste sous la forme <list-aggregate> (équivalent à <items group-by="*")
- Attribut @aggregate-function (équivalent à @grouping-function). A utiliser sur les éléments <field>.
- Indentation supprimée, permettant un gain de taille de fichier de l'ordre de 30% à 40%
  Cependant, un paramètre technique DB.ind permet de forcer une indentation en le renseignant avec des espaces ou des tabulations.

1.57 - 23/12/2023

Nouveautés :

- Soustractions et divisions de champs composés :
  function="sub()" : soustraction
  function="div()" : division
  Exemple :

```
<field name="total" function="sub()">
    <field id="valeur_initiale"/>
    <field id="valeur_a_retrancher"/>
</field>
```

- Remplacement des variables ${xxxxx} lors de l'utilisation de l'attribut @math
  Exemple : `<field id="valeur_initiale" math=".+ ${valeur_a_ajouter}"`

1.56 - 21/12/2023

Nouveautés :

- Ajout d'options de formatage à la fonctionnalité @split :
  Alignement du texte :
  split="*longueur* , **justify**" → justification du texte (valeur par défaut)
  split="*longueur* , **no-justify**" → désactivation de la justification
  Saut de ligne :
  split="*longueur* , **break-on-punctuation**" → les caractères de ponctuation peuvent générer un saut de ligne automatique, même en cas de présence d'espaces sécables dans le texte.
  Ces caractères sont les suivants : **- / ! ? ( ) { } : ;**
  Cela permet de passer un peu moins rapidement à la ligne et donc d'économiser de l'espace vertical sur le document en sortie.

  Exemple :
  Texte en entrée : `D. HOUSSE 12M123456/COTON BLANC` :
  Résultat avec @**split="20 , no-justify"** :

```
D. HOUSSE
12M123456/COTON
BLANC
```

  Résultat avec @**split="20 , no-justify , break-on-punctuation"** :

```
D. HOUSSE 12M123456/
COTON BLANC
```

Évolutions :

- @split : prise en charge des tabulations éventuellement présentes dans le texte en les remplaçant par 8 espaces

Corrections :

- Correction du nom des champs lors de l'utilisation de l'attribut @split sur des éléments <table> de type « address ». Pour rappel, avec la création d'un tableau de type "address" , tous les champs doivent être nommés "value"
- Suppression de lignes blanches intempestives lors de l'utilisation de la fonction @split dans le cas où le texte en entrée comporte des sauts de ligne.
- Amélioration des performances sur le traitement de gros fichiers XML.

1.55 - 17/08/2023

Nouveautés :

- Paragraphes dynamiques <paragraph> : possibilité d'utiliser des balises **{{col:*nom_champ*}}** dans le texte afin de présenter une partie de ce texte sous forme de colonnes.
  Cette balise a pour effet de répartir le texte dans différents champs pour le Designer.

Évolutions :

- Attribut @cut **avec l'option « tail »** : la longueur spécifiée permet de récupérer les x caractères vers la gauche à partir du point de départ.

  Exemples pour récupérer les derniers caractères d'une chaine "123456789" :
  cut="1-3,tail" → "789" (caractères entre les positions 1 à 3 en partant de la fin)
  cut="3-1,tail" → "789" (caractères entre les positions 3 à 1 en partant de la fin)
  cut="1,3,tail" → "789" (longueur de 3 caractères **vers la gauche** à partir du dernier caractère)
  cut="3,3,tail" → "567" (longueur de 3 caractères **vers la gauche** à partir du 3ème en partant de la fin)
- Le fichier XML de paramétrage est maintenant facultatif dans lors de l'invocation du Connect XML4DESIGN (-param:DB.PARAM_XML).
- Paragraphes dynamiques <paragraph> :
  - Possibilité de définir le type d'objet à utiliser dans le Designer pour mettre en forme le paragraphe
    <paragraph **object-type="group"**> : utilisation d'un groupe (valeur par défaut)
    <paragraph **object-type="text-object"**> : utilisation d'un objet « texte »
  - Possibilité de spécifier le nom du fichier texte en se basant sur plusieurs champs, sans passer par des variables intermédiaires :

    Exemple : <paragraph file="**field(*mon_champ_1*)_field(*mon_champ_2*)**.txt"
  - Ajout d'un commentaire dans le XML en sortie, avec le chemin complet du fichier réellement utilisé (pour débogage)

1.54 - 29/05/2023

Nouveautés :

- Option "**tail**" sur l'attribut @cut permettant de conserver uniquement les derniers caractères d'un champ. De plus, en cas d'usage de la longueur, cette dernière peut maintenant être négative pour récupérer les caractères situés à gauche de la position spécifiée. Dans tous les cas la chaine récupérée n'est jamais inversée (l'ordre des caractères n'est pas modifié).

  Syntaxe : cut="*startPos*-*endPos*[,*origin*]" ou cut="startPos,length[,origin]"

  startPos : position du caractère de départ
  endPos : position du caractère d'arrivée
  length : longueur de la chaine à récupérer. Si positive : vers la droite , si négative : vers la gauche
  origin (facultatif) : origine du positionnement : head (début) ou tail (fin)

  Le positionnement est toujours numéroté à partir de 1

  Exemples pour récupérer les derniers caractères d'une chaine "123456789" :
  - cut="1-3,tail" → "789" (caractères entre les positions 1 à 3 en partant de la fin)
  - cut="3-1,tail" → "789" (caractères entre les positions 3 à 1 en partant de la fin)
  - cut="3,3,tail" → "789" (longueur de 3 caractères vers la droite à partir du 3ème en partant de la fin)
  - cut="1,-3,tail" → "789" (longueur de 3 caractères vers la gauche à partir du dernier caractère)

  Par symétrie, l'option "head" (par défaut) permet de partir de la gauche.

> **Attention :**
> **Modification du comportement de l'option "tail" à partir de la version 1.56 du 21/12/2023**

1.53 - 29/04/2023

Nouveautés :

- Attribut **@keep-empty-if-zero** au niveau <field> permettant de ne pas afficher la valeur du champ si celle-ci est numériquement égale à zéro
  valeur : "true" | *"false"*

Evolutions :

- Paragraph : les espaces en début de chacune des lignes du fichier texte sont systématiquement figés (sous formes d'espaces insécables) afin de **respecter le souhait d'indentation de l'utilisateur**.
- Protection de tous les champs numériques, date, prefix et suffix afin qu'il ne puisse pas y avoir de saut de ligne en plein milieu même s'ils contiennent des espaces (exemple : ne pas générer de saut de ligne entre le 1 et le 2 du montant "1 234,56 €")
- Si le <doc> n'a pas de nom, l'élément <page> résultant sera nommé "normal" : *<page name="normal">*

Corrections :

- Correction de bug concernant l'usage de @columns sans <line> explicite
- Correction de bug en cas d'utilisation conjointe de label-sheet ET @indivisible="true"
- Balise </line> intempestif lors de certaines utilisations du mode "label-sheet"

1.52 - 27/04/2023

- <paragraph> : les variables dans les fichiers “texte” doivent être présentes sous la forme [[xxxxx]] ou {{field:xxxxx}}
- <paragraph> : en cas de présence de tabulations dans les fichiers “texte” , ces dernières sont automatiquement remplacées par 8 espaces (7 espaces insécables et 1 espace classique)
- <paragraph> : prise en charge de l’attribut @copy-number permettant de conditionner le paragraphe selon le numéro d’exemplaire du document (en cas d’utilisation de @nb-copies permettant de sortir un document en plusieurs exemplaires)
- Champs composés : nouvelle fonction "first-not-empty()" permettant de récupérer uniquement la valeur du premier champ non vide trouvé.
- @split-line-replicate : ajout de la possibilité d’utiliser le mot clé "first" afin de faire écho au mot clé "last". Ainsi **split-line-replicate="first"** est équivalent à **split-line-replicate="1"**

1.51 - 26/04/2023

- Nouveau : éléments <paragraph> permettant d’inclure du contenu provenant de fichiers textes. Idéal pour confectionner des courriers, avec des variables et y inclure des listes
- Fix : anomalie sur les champs “split” dans les listes dans le cas où il n’y avait pas de <line> explicite
- Amélioration : divers optimisations

1.50 - 31/03/2023

- Nouveau : attribut **@name** sur les éléments <doc>, permettant de personnaliser le nom des éléments <page> créées dans le XML Designer.
- Fix : correction de lecture de données sur les champs composés hors <line> pour les listes de niveau 2 et supérieurs

1.49 - 14/02/2023

- Nouveau : attribut **@item-type** (niveau <line> , <field> , <cond> , <var> et <list>) permettant de conditionner l'élément en fonction du type d’item en cours (attribut @type au niveau <item> dans le fichier XML data.
- Optimisation du temps de traitement lors de l’usage de l’attribut @if
- Fix : l’attribut @item-number n'était pas bien pris en charge sur les éléments <line>

1.48 - 06/02/2023

- Nouveau : attribut **@nb-integer** (niveau <field>) permettant de forcer un nombre minimum d'entier sur un champ numérique
- Nouveau : attributs **@replace-in et @replace-out** (niveau <field>) permettant de remplacer une sous chaine de caractères par une autre sur un champ (action effectuée en tout début de manipulation du champ, avant le @cut)
- Fix : l'attribut @nb-copies est maintenant bien pris en charge au niveau <param> si le niveau <doc> n'est pas présent

1.47 - 16/01/2023

- Prise en charge du nombre de copies permettant de spécifier le nombre d'exemplaires d'un document :
  - Attribut **@nb-copies** utilisable au niveau <param> ou au niveau <doc>
  - paramètre d'exécution ***DB.nb-copies***=*xxxxxxx Exemples :* @nb-copies="3" ou @nb-copies="field(monchamp)" ou @nb-copies="global-field(monchamp)"
  - Conditionnement des éléments avec l'attribut **@copy-number** (fonctionnement identique à item-number)
  - Ajout de variables automatiques : **${MAPPING_NB_COPIES} , ${MAPPING_COPY_NUMBER}**
- Fix : l'attribut @if n'était pas pris en charge pour les élément <cond> en dehors des listes (à la racine de <doc>)

1.46 - 30/11/2022

- Fix : suppression d'un chemin de fichier en dur sur le tableau tab_group_list1 provoquant des erreurs lors des regroupements du type group-by="*"

1.45 - 07/11/2022

- Suppression de l'information encoding="UCS-2" dans l'entête. En effet, le BOM "FFFE" est suffisant pour spécifier au parseur que le fichier est encodé en UCS-2. Et en cas de conversion en UTF-8, il n'est plus nécessaire de modifier cette valeur, sachant que par défaut, les XML sont considérés comme étant encodés en UTF-8.

1.44 - 02/11/2022

- Evol : ajout d'un paramètre d'exécution **"-param:DB.comment=false"** permettant de désactiver les commentaires dans le flux XML en sortie, afin d'alléger ce dernier en production
- Evol : pour les tableaux de type adresse (<table ***type="address"***>), un nouvel attribut @**keep-blank-line="true"** permet de forcer la conservation des lignes vides. Valeur par défaut : "false"
- Evol : les paramètres techniques du maprunrpt **MAPPING_DATASTREAM, MAPPING_WIDTHPAGE et MAPPING_MAXWHILE** ne doivent plus être renseignés dans la ligne de commande. Ils sont renseignés automatiquement en interne, dans le code du Connect.
- Fix : prise en charge des éléments <var> et <cond> à leur emplacement réel (avant tous les éléments hors <line> étaient évalués d'un seul bloc, même s'ils n'étaient pas contigus)
- Fix : les listes sont maintenant limitées à 9.999.999 items au lieu de 99.999 auparavant
- Fix : dans certains cas le XML Designer pouvait comporter des <line> imbriquées

1.43 - 09/09/2022

- Fix : un bug empêchait de pouvoir utiliser les éléments <cond> directement à la racine d'un document, c'est-à-dire en dehors des <list>

1.42 - 05/09/2022

- Evol : ne plus créer implicitement des éléments <line> dans le XML Designer s'il y a uniquement présence d'éléments <var> ou <cond> sans aucun champ <field> en dehors d'une <line>. Ce qui permet de faciliter les cumuls en parcourant tous les items d'une liste, sans pour autant se retrouver avec plein de <line> inutiles qu'il est nécessaire de conditionner dans le Designer !
- Nouveau : implémentation des conditions avec les éléments <cond> et les attributs @if

1.41 - 13/08/2022

- Nouveau : création d’un flux XML "Designer" spécifique aux planches d'étiquettes (suffixe "_x" sur les champs issus de listes à la place de la création d’un élément <group>)
  Paramètres :
  - DB.output-type=label-sheet , ou attribut @output-type="label-sheet" sur les éléments <param> et/ou <doc>
  - DB.labels-per-sheet, ou attribut @labels-per-sheet sur les éléments <param> et/ou <doc> : nombre d'étiquettes par page
- Nouveau : épurer le XML "Designer" de ses champs <field> vides (uniquement les champs issus de listes)
  Paramètre : DB.display-empty-fields ou attribut @display-empty-fields sur l'élément <param>
  Valeurs attendues : true|false
  Valeur par défaut : true
  Si "false" : les champs vides (sans valeurs) dans les listes ne sont pas créés dans le flux de sortie XML "Designer"

1.40 - 12/08/2022

- Version rejetée

1.39 - 27/07/2022

- Fix : correction requête XML pour accéder aux données des champs composés pour les listes de niveau 4 et 5

1.38 - 05/07/2022

- Nouveau : Ajout possibilité de définir le nom d’une <line> du XML "Designer" grâce à la valeur de l’un des champs de l’item courant.
  Syntaxe : <line name="field(myField)">…. avec myField le nom d’un champ

1.37 - 11/05/2022

- Nouveau : Prise en charge d’un 5ème niveau de liste

1.36 - 20/04/2022

- Fix : erreurs de prises en charge de variables dans les champs composés des listes.

1.35 - 20/04/2022

- Nouveau : prise en charge du @cut permettant de faire une extraction partielle d’un champ :
  - cut="3,7" → à partir du 3ème caractère sur une longueur de 7
  - cut="3-9" → à partir du 3ème caractère jusqu’au 9 ème caractère
  - cut="integer-part" → sur un champ numérique, permet de récupérer uniquement la partie entière
  - cut="decimal-part" → sur un champ numérique, permet de récupérer uniquement la partie décimale
- Prise en compte des attributs sur les variables passées en valeur sur des champs :
  <field name="mnt" nb-decimal="2">${mymnt}</field>
  --> Uniquement si la valeur text() est la seule source de donnée. Donc si @id ou @from est spécifié, la valeur est toujours ajoutée simplement en préfixe sans être modifiée (rétro-compatibilité)

1.34 - 30/03/2022

- Fix : possibilité usage simultané de @same-line-as-parent et @culumns

1.33 - 22/09/2021

- Nouveau : création de l'attribut @split-line-replicate permettant de répliquer certains champs sur les lignes automatiquement ajoutées lors de l'utilisation de l'attribut @split sur un champ. Pour rappel, par défaut, seul le champ comportant l'attribut @split est présent dans les <line> générées automatiquement.
  Valeurs possibles : "all", "last", "N" avec N entier positif
- Fix : ne plus lire les <field> n'ayant pas d'identifiant (@id) dans le fichier XML de données

1.32 - 07/07/2021

- Fix : erreur d'interprétation de <list-footer> sur les listes de niveau 2 (ajout init compteur d'item)

1.31 - 04/07/2021

- Prise en charge des éléments <var> composés de sous noeuds <field>

1.30 - 25/05/2021

- Fix: pb en cas d'utilisation de l'attribut "columns" sur les listes de niveau 1. Le suffixe permettant de différencier les champs des différentes colonnes n'était pas appliqué sur la dernière colonne.
- Fix : dans le cas où une liste comportait un seul item, en cas d'utilisation du mode "columns", le dernier élément <line> n'était pas fermé
- Fix : les éléments <items> permettant d'agréger tous les enregistrements d'une liste (group-by="*") doivent être appliqués même si la liste ne contient aucun enregistrement
- Nouveau : implémentation de l'attribut @apply-condition sur les éléments <list-header> , <list-footer> ou en cas d'utilisation de l'attribut @group-by="*" sur un élément <items>, avec valeur par défaut="always".
  Valeurs possibles :
  - "always"
  - "filled-list" → uniquement si la liste contient au moins un item
  - "empty-list" → uniquement si la liste est vide

1.29 - 14/05/2021

- Fix: en cas d'utilisation de l'attribut @indentation conjointement au @split, les lignes "suivantes" (2, 3 4…) comportaient moins de caractères qu'attendu.
- En cas d'utilisation du @split sur un champ, si ce dernier comporte des retours chariots (CR, LF, CRLF), ces derniers seront pris en compte pour ajouter un saut de ligne même si la longueur spécifiée par l'attribut @split n'est pas atteinte.

1.28 - 15/04/2021

- Fix: ajout réinit DB.line_column_number=0 en début de traitement des listes suite à l'apparition intempestive de suffixes "_1" , "_2" sur certains champs.

1.27 - 13/04/2021

- Création/modification de variables <var> au sein des listes

1.26 - 24/03/2021

- Elément <var> (déclaration et utilisation sous la forme ${myVar}). Déclaration limitée à la racine du niveau <doc>
- @from et @where sur les listes, les champs à la racine du document et sur l'élément <var>

1.25 - 13/03/2021

- En cas d'utilisation de l'attribut @columns , le suffixe avec le numéro de colonne doit être présent même si la zone ne doit pas être présente.

1.24 - 08/03/2021

- Numérotation de TOUTES les lignes dès la première à partir du moment où il y a un split sur un champ.
- Conserver et laisser vides les champs non trouvés dans le flux de données ainsi que ceux sur lesquels il y a une condition (item-number, if....)

1.23 - 03/03/2021

- Ajout d'un champ MAPPING_SPLIT_LINE sur chaque ligne créée automatiquement suite à l'usage de l'attribut @split

1.22 - 02/03/2021

- Fix case="first-upper"
- Langue "fr" pour les date sous linux

1.21 - 01/03/2021

- Fix: bug sur les listes de niveau 2 qui prenaient en comptes les sauts de lignes du XML de paramétrage

1.20 - 01/03/2021

- Application des attributs @prefix et @suffix uniquement si la valeur finale du champ n'est pas vide

1.19 - 25/03/2021

- Prise en charge de function="count()" sur les <field> se trouvant dans des <table>

1.18 - 19/02/2021

- Champ considéré implicitement comme numérique (@data-type="number") en cas d'utilisation de @math

1.17 - 15/02/2021

- @math : application d'une opération arithmétique sur un champ.
  Syntaxe limitée à :
  - math="./123"
  - math="*123"
  - math=".+123"
  - math=".-123"
  - math=".%123"

  avec "123" une valeur numérique

1.16 - 14/02/2021

- @nb-decimal avec possibilité de renseigner un nombre min et max de chiffres après la virgule

1.15 - 12/02/2021

- @keep-empty-if-void
- @prefix
- @suffix

1.14 - 07/02/2021

- Prise en charge des valeur NaN comme étant également à zéro lors des calculs de sommes
- Prise en compte du formatage numérique par défaut pour les résultats de calculs

1.13 - 05/02/2021

- Ajout info version en début de fichier XML "Designer"
- Ajout title="DEPRECATED" sur les éléments qui ne seront pas repris tels quels dans la version C++

1.12 - 28/01/2021

- modification du formatage de dates en sortie (date-out) en phase avec les nouvelles specs sur le sujet

1.11 - 21/01/2021

- list-header et list-footer
- attribut @same-line-as-parent
- prise en charge d'un 4ème niveau de <list>

1.10 - 11/12/2020

- Implémentation uniquement sur <table> :
  - sum() et avg() sur collection de résultat "from" , ainsi que "where" avec un seul filtre

1.9 - 09/09/2020

- amélioration du "split" avec prise en charge des caractères de ponctuation comme ? ! , ; -
- xfrom = from (x devenu inutile)
- élément <items> facultatif et possibilité d'utiliser ses attributs directement au niveau de l'élément <list>
- xfrom avec global-data() , current-doc() , parent-list()
- conversion de dates (date-in , date-out, culture) sous la forme [DD] [MM] [YYYY] etc…
- case (upper , lower , first-upper)
- option permettant de cumuler des champs sur chaque ligne

1.8 - 14/08/2020

- grouping-function → implémentation de avg() et count() (résultat implicitement numérique)
- si aucun élément n'est trouvé dans un élément <list> alors tout est récupéré pour cette liste
- Simplification param basic : <param></param> est suffisant pour tout récupérer et faire un XML Designer sans particularité

1.7 - 07/08/2020

1.6 - 14/05/2020

1.5 - 30/03/2020

1.4 - 19/03/2020

1.3 - 19/03/2020

1.2 - 18/03/2020

1.1 - 17/03/2020