Documentation EMu

L’ajout de données en utilisant l’outil Importer

Version Document 1.0

Version EMu 4.0

Contents

SECTION 1

Vue d'ensemble

KE EMu 3.2.01 a ajouté la possibilité d'importer des enregistrements spécifiés soit en XML (eXtensible Markup Language) ou en CSV (Comma Separated Values). Le nouvel outil Importer permettait la création d'enregistrements, y compris la création d'enregistrements de références. Il est donc possible d'importer un nouvel enregistrement Catalogue et de créer un enregistrement Personnes/Org. pour le champ *Créateur* si l'enregistrement Personnes/Org. n'existe pas encore. L'outil Importer fournit aussi un mécanisme de mise à jour des enregistrements. En utilisant la fonction de mise à jour il est possible de remplacer le contenu d'un champ existant avec de nouvelles données. Malheureusement, le contenu complet du champ est remplacé par les données importées.

Dans certains cas il peut être utile de mettre à jour une liste de valeurs en :

- ajoutant des valeurs après une position donnée ou après la dernière valeur.
- ajoutant des valeurs avant une position donnée ou avant la première valeur.
- remplaçant des valeurs à une position donnée.

EMu 4.0.02 étend le mécanisme d'Importation, permettant aux valeurs de tables et de tables imbriquées d'être mises à jour plutôt que d'être remplacées. Il est désormais possible d'ajouter après, d'ajouter avant et de remplacer des valeurs d'une position donnée avec des données importées.

Un nouveau mécanisme group permet de relier entre elles des colonnes séparées lors de l'ajout de données, s'assurant que toutes les valeurs sont placées dans la même rangée pour chaque colonne. Par exemple lors de la mise à jour d'un enregistrement Catalogue, vous voulez peut être que les valeurs *Créateur* et *Date de création* apparaissent dans la même rangée, puisque chaque élément d'information est lié à l'autre.

Dans la section suivante, nous voyons les extensions ajoutées à l'outil Importer pour le support de l'adjonction de données pour les sources de données XML et CSV.

Veuillez noter que les séparateurs utilisés dans Microsoft Excel pour la création de fichiers CSV ne sont pas standardisés (virgules, points-virgules, etc.). Pour utiliser la fonction Importer d'EMu, il est nécessaire de changer les Options régionales et linguistiques de votre ordinateur à *Anglais* dans le panneau de configuration lors de la sauvegarde du fichier CSV. (Les donnés peuvent être saisies en utilisant vos Options régionales habituelles.)

SECTION 2

Adjonction de données

La fonctionnalité Importer d'EMu permet d'utiliser XML ou CSV pour définir des enregistrements à importer. Le format XML utilisé est le même que celui produit par l'outil de création de rapports d'EMu, alors que CSV est une norme de facto pour l'échange de données. Considérer le XML ci-dessous :

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple>
                <atom>Map Number 2314</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

ou le CSV équivalent :

irn	LatSource_tab(1)
5000432	Map Number 2314

Lors de l'importation, toutes les valeurs dans la colonne *LatSource_tab* seront remplacées par la valeur `Map Number 2314`. Pour fournir un support de mise à jour des rangées dans une table, un nouvel attribut rangée (row) a été ajouté à l'élément `<tuple>` pour XML, permettant de spécifier une position de rangée. L'attribut row fournit un mécanisme pour indiquer quel type de mise à jour doit être appliqué et quelle rangée est affectée. Le format de l'attribut est :

`<tuple row='`*value*`'>`

Pour une source de données CSV, le numéro de rangée est entre parenthèses. CSV permet à deux formats d'être utilisés pour spécifier l'attribut row :

`(row='`*value*`')`

ou la forme plus courte :

`(`*value*`)`

Bien que les guillemets puissent être utilisés pour encadrer la valeur (*value*) de row, les apostrophes sont recommandées pour les fichiers CSV car le format CSV utilise des guillemets pour encadrer des valeurs de données. Lors de la sauvegarde des fichiers CSV dans Excel, une sortie incorrecte sera produite si les guillemets sont utilisés lorsque vous spécifiez la valeur de la rangée.

La valeur de l'attribut row peut être :

nnn	où **nnn** est un numéro de rangée. Le numéro indique la position de la rangée dans la liste des valeurs à modifier. La première rangée est numérotée 1. En gros, ce paramètre remplace la valeur actuelle à cette position.
+	indique que la position de la rangée est après la dernière valeur dans la table. Toutes les données seront ajoutées après la liste de valeurs.
-	indique que la position de la rangée est avant la première valeur dans la table. Toutes les données seront ajoutées au début de la liste avec les valeurs existantes déplacées vers le bas.
=	indique que la position de la rangée est rangée une. Toutes les données ajoutées remplaceront les valeurs existantes à cette position.
nnn+	ajoute toutes les données après rangée numéro *nnn*.
nnn-	ajoute toutes les données avant rangée numéro *nnn*.
nnn=	remplace toutes les données à la rangée numéro *nnn*.

Dans tous les cas, si le nouveau numéro de rangée n'existe pas, il est créé. Par exemple si le paramètre row est :

```
<tuple row='12+'>
```

ou pour CSV :

`(row='12+')` ou `(12+)`

et il n'y a pas douze valeurs dans la table, la table sera agrandie pour contenir douze valeurs (avec des rangées vides) et la nouvelle valeur ajoutée à la fin, créant une treizième rangée. Si l'attribut row n'est pas défini, le comportement par défaut consiste à ajouter à la fin de la table.

Afin d'assurer la compatibilité avec l'outil Importer existant et de permettre de remplacer toutes les valeurs d'une table, le premier <tuple> d'une table pour XML, ou le premier spécificateur de rangée pour CSV, définit si les valeurs d'une table sont mises à jour ou si toutes les valeurs sont remplacées. Si l'attribut row n'est pas défini dans XML, ou juste un numéro de rangée est spécifié, alors le contenu de la colonne sera effacé et les valeurs importées seront ajoutées. Si un attribut row est fourni et contient opérateur de mise à jour (c'est à dire +, - ou = avec ou sans numéro de rangée), le contenu de la table sera mis à jour, avec les valeurs existantes conservées.

La section suivante contient des exemples sur comment remplacer ou mettre à jour le contenu d'une table en utilisant le nouvel attribut row. Ces exemples donnent des solutions XML et CSV.

Mise à jour des tables

Les exemples ci-dessous montrent comment l'attribut row peut être utilisé pour remplacer ou mettre à jour le contenu d'une table.

Exemple un – Remplacer le contenu d'une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple>
                <atom>new source 1</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(1)
5000432	new source 1

Comme la valeur de l'attribut row dans la colonne *LatSource_tab* n'a pas d'opérateur de mise à jour (c'est à dire +, - ou =), le contenu de la colonne est effacé avant d'ajouter les nouvelles données. Les tables ci-dessous montrent les données avant et après l'importation :

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	new source 1

Exemple deux - Remplacer le contenu d'une table et sauter des rangées

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="1">
                <atom>new source 1</atom>
            </tuple>
            <tuple row="3">
                <atom>new source 3</atom>
            </tuple>
            <tuple row="5">
                <atom>new source 5</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(1)	LatSource_tab(3)	LatSource_tab(5)
5000432	new source 1	new source 3	new source 5

Une fois encore, comme le premier attribut row ne contient pas d'opérateur de mise à jour, le contenu de *LatSource_tab* sera effacé avant que les données importées soient ajoutées. L'utilisation de l'attribut row permet de définir les positions spécifiques des rangées. L'exemple ci-dessous montre les données avant et après l'importation :

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	new source 1
2	
3	new source 3
4	
5	new source 5

Exemple trois – Ajouter une valeur à la fin d'une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+">
                <atom>append source 1</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(+)
5000432	append source 1

Comme l'attribut row contient un opérateur de mise à jour, +, le contenu existant du champ *LatSource_tab* est conservé et la nouvelle valeur ajoutée à la fin de la table :

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	source 1
2	source 2
3	source 3
4	source 4
5	append source 1

Exemple quatre - Ajouter des valeurs multiples à la fin d'une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+">
                <atom>append source 1</atom>
            </tuple>
            <tuple row="+">
                <atom>append source 2</atom>
            </tuple>
            <tuple row="+">
                <atom>append source 3</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(+)	LatSource_tab(+)	LatSource_tab(+)
5000432	append source 1	append source 2	append source 3

Comme le premier attribut row contient un opérateur de mise à jour (l'opérateur +), les données importées mettent à jour les valeurs existantes de la table. Remarquer l'utilisation de l'opérateur de mise à jour pour chaque rangée à ajouter:

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	source 1
2	source 2
3	source 3
4	source 4
5	append source 1
6	append source 2
7	append source 3

Exemple cinq - Ajouter une valeur au début d'une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="-">
                <atom>prepend source 1</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(-)
5000432	prepend source 1

Insérer une valeur dans la première position de la table est semblable à ajouter une valeur à la fin, sauf que l'opérateur (-) est utilisé. Quand la valeur est insérée, les valeurs existantes de la table sont déplacées vers la bas :

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	prepend source 1
2	source 1
3	source 2
4	source 3
5	source 4

Example six – Remplacer les valeurs dans une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="2=">
                <atom>replace source 1</atom>
            </tuple>
            <tuple row="3=">
                <atom>replace source 2</atom>
            </tuple>
            <tuple row="6=">
                <atom>replace source 3</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(2=)	LatSource_tab(3=)	LatSource_tab(6=)
5000432	replace source 1	replace source 2	replace source 3

L'opérateur = est utilisé pour remplacer le contenu d'une rangée dans une table. Si la rangée n'existe pas, elle est créée avec la valeur appropriée :

Avant

1	source 1
2	source 2
3	source 3
4	source 4

Après

1	source 1
2	replace source 1
3	replace source 2
4	source 4
5	
6	replace source 3

Exemple sept - Ajouter une référence à la fin d'une table

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatDeterminedByRef_tab'>
            <tuple row="+">
                <atom name="NamFirst">firstname</atom>
                <atom name="NamLast">lastname</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatDeterminedByRef_tab(+).NamFirst	LatDeterminedByRef_tab(+).NamLast
5000432	firstname	lastname

Tous les opérateurs de mise à jour disponibles pour les tables de valeurs peuvent être utilisés pour les tables de références. Pour les sources de données CSV, les colonnes portant le même nom et attribut row sont traités comme une mise à jour. Par exemple les deux attributs row du CSV ci-dessus utilisent l'opérateur de mise à jour (+). Cela ne résulte pas en deux nouvelles rangées mais plutôt la même rangée est utilisée pour contenir la référence :

Avant

1	reference 1
2	reference 2
3	reference 3
4	reference 4

Après

1	reference 1
2	reference 2
3	reference 3
4	reference 4
5	append reference 1

Mise à jour des tables imbriquées

Le mécanisme de mise à jour des tables imbriquées, c'est à dire de tables à l'intérieur de tables, est très semblable à la mise à jour d'une table simple. La seule différence est que les tables imbriquées ont un numéro de rangée extérieure et un numéro de rangée intérieure. Chaque rangée extérieure contient une table de valeurs de rangée intérieure :

Rangée extérieure	Rangée intérieure	Données
1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Comme vous pouvez le voir dans le tableau ci-dessus, rangée extérieure une contient une table avec trois valeurs, alors que rangée extérieure deux a une valeur et rangée extérieure trois a deux valeurs. La représentation XML du tableau ci-dessus est :

```
<table table='ecollectionevents'>
    <tuple>
        <table name='LatComment_nesttab'>
            <tuple>
                <table>
                    <tuple>
                        <atom>opinion 1, comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>opinion 1, comment 2</atom>
                    </tuple>
                    <tuple>
                        <atom>opinion 1, comment 3</atom>
                    </tuple>
                </table>
            </tuple>
            <tuple>
                <table>
                    <tuple>
                        <atom>opinion 2, comment 1</atom>
                    </tuple>
                </table>
            </tuple>
            <tuple>
                <table>
                    <tuple>
                        <atom>opinion 3, comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>opinion 3, comment 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

La table imbriquée *LatComment_nesttab* commence avec l'élément `<table name='LatComment_nesttab'>`. Chaque rangée extérieure est encadrée par un élément `<tuple>`. Les rangées extérieures sont en vert. A l'intérieur de chaque rangée extérieure se trouve une table de rangées intérieures. Les rangées intérieures sont encadrées par les éléments `<tuple>` rouges. Lors de la mise à jour de tables imbriquées, les éléments `<tuple>` des rangées extérieures et intérieures peuvent avoir des attributs row. Il est donc possible d'ajouter avant/d'ajouter après/de remplacer des rangées extérieures et/ou intérieures.

La représentation CSV équivalente est montrée ci-dessous. Le tableau a été tourné sur le côté pour aisance de visualisation. Le nom des colonnes devrait apparaitre sur la première ligne plutôt que dans la première colonne :

irn	5000432
LatComment_nesttab(1:1)	opinion 1, comment 1
LatComment_nesttab(1:2)	opinion 1, comment 2
LatComment_nesttab(1:3)	opinion 1, comment 3
LatComment_nesttab(2:1)	opinion 2, comment 1
LatComment_nesttab(3:1)	opinion 3, comment 1
LatComment_nesttab(3:2)	opinion 3, comment 2

Les rangées extérieures et intérieures des tables imbriquées en format CSV sont enregistrées après le nom de la colonne entre parenthèses séparées par deux-points. La rangée extérieure est montrée en vert et la rangée intérieure en rouge. L'exemple ci-dessus utilise la forme courte de l'attribut row. La forme longue ressemblerait à ceci :

```
LatComment_nesttab(row='1':row='1')
```

Comme pour XML, les rangées extérieures et/ou intérieures peuvent contenir des operateurs de mise à jour. Les exemples suivants détaillent quelques utilisations courantes.

Exemple un – Remplacer le contenu d'une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple>
                <table>
                    <tuple>
                        <atom>new opinion 1, new comment 1</atom>
                    </tuple>
                </table>
            </tuple>
            <tuple>
                <table>
                    <tuple>
                        <atom>new opinion 2, new comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>new opinion 2, new comment 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_ nesttab(1:1)	LatComment_ nesttab(2:1)	LatComment_ nesttab(2:2)
5000432	new opinion 1, new comment 1	new opinion 2, new comment 1	new opinion 2, new comment 2

Comme la première rangée extérieure et la première rangée intérieure ne contiennent pas d'attribut row, le contenu actuel de la colonne *LatComments_nesttab* sera effacé :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	new opinion 1, new comment 1
2	1	new opinion 2, new comment 1
	2	new opinion 2, new comment 2

Exemple deux – Ajouter une rangée extérieure à une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple row="+">
                <table>
                    <tuple>
                        <atom>append opinion 1, comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>append opinion 1, comment 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_nesttab(+:1)	LatComment_nesttab(+:2)
5000432	append opinion 1, comment 1	append opinion 1, comment 2

La première rangée extérieure a un attribut row avec une valeur de +, indiquant qu'une rangée extérieure doit être ajoutée après les données existantes. La table intérieure contient les valeurs à ajouter :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2
4	1	append opinion 1, comment 1
	2	append opinion 1, comment 2

Exemple trois – Ajouter des valeurs à une table intérieure dans une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple row="2">
                <table>
                    <tuple row="+">
                        <atom>opinion 2, append comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>opinion 2, append comment 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_nesttab(2:+)	LatComment_nesttab(2:+)
5000432	opinion 2, append comment 1	opinion 2, append comment 2

La rangée extérieure contient un numéro de rangée fixe alors que la première rangée intérieure contient un opérateur de mise à jour, l'operateur (+). La combinaison indique que les valeurs sont à ajouter à la fin de la deuxième rangée extérieure :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
	2	opinion 2, append comment 1
	3	opinion 2, append comment 2
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Exemple quatre - Ajouter une rangée extérieure avant une autre dans une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple row="2-">
                <table>
                    <tuple>
                        <atom>prepend opinion 2, comment 1</atom>
                    </tuple>
                    <tuple>
                        <atom>prepend opinion 2, comment 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_nesttab(2-:1)	LatComment_nesttab(2-:2)
5000432	prepend opinion 2, comment 1	prepend opinion 2, comment 2

L'attribut de rangée extérieure à valeur de `2-` ajoutera une rangée extérieure avant la rangée deux et déplacera la rangée deux existante et les rangées suivantes vers le bas. La table intérieure contient les valeurs à ajouter :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	prepend opinion 2, comment 1
	2	prepend opinion 2, comment 2
3	1	opinion 2, comment 1
4	1	opinion 3, comment 1
	2	opinion 3, comment 2

Exemple cinq – Remplacer une rangée extérieure dans une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple row="2=">
                <table>
                    <tuple>
                        <atom>replace opinion 2, comment 1</atom>
                    </tuple>
                    <tuple row="3">
                        <atom>replace opinion 2, comment 3</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_nesttab(2=:1)	LatComment_nesttab(2=:3)
5000432	replace opinion 2, comment 1	replace opinion 2, comment 3

L'attribut de rangée `2=` sur la rangée extérieure indique que la rangée deux doit être remplacée. Notez l'utilisation de l'attribut row dans la table intérieure pour sauter la rangée deux (créant ainsi une valeur vide) :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	replace opinion 2, comment 1
	2	
	3	replace opinion 2, comment 3
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Exemple six - Remplacer une rangée extérieure dans une table imbriquée

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatComment_nesttab'>
            <tuple row="3">
                <table>
                    <tuple row="1=">
                        <atom>opinion 3, replace comment 1</atom>
                    </tuple>
                    <tuple row="+">
                        <atom>opinion 3, append comment 1</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatComment_nesttab(3:1=)	LatComment_nesttab(3:+)
5000432	opinion 3, replace comment 1	opinion 3, append comment 1

L'attribut row de la rangée extérieure indique que la rangée trois doit être modifiée, et le premier attribut row de la rangée intérieure indique que la rangée une doit être remplacée. Le deuxième attribut row de rangée intérieure ajoutera une valeur à la fin de la table intérieure :

Avant

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, comment 1
	2	opinion 3, comment 2

Après

1	1	opinion 1, comment 1
	2	opinion 1, comment 2
	3	opinion 1, comment 3
2	1	opinion 2, comment 1
3	1	opinion 3, replace comment 1
	2	opinion 3, comment 2
	3	opinion 3, append comment 1

Groupes

La capacité d'ajouter des données à une liste existante permet d'ajouter de nouvelles valeurs à des colonnes au fil du temps. Un problème qui peut se poser est la nécessité d'ajouter des valeurs à plus d'une colonne, en veillant à ce que les valeurs soient ajoutées à la même rangée. Considérer la mise à jour suivante :

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+">
                <atom>append source 1</atom>
            </tuple>
        </table>
        <table name='LatLatLongDetermination_tab'>
            <tuple row="+">
                <atom>append determination 1</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(+)	LatLatLongDetermination_tab(+)
5000432	append source 1	append determination 1

ayant pour résultat :

Avant

1	source 1	determination 1
2	source 2	determination 2
3	source 3	

Après

1	source 1	determination 1
2	source 2	determination 2
3	source 3	append determination 1
4	append source 1	

Ces données ne sont presque certainement pas ce qui était envisagé. Les deux colonnes, *LatSource_tab* et *LatLatLongDetermination_tab* sont liées et apparaissent dans une seule grille. En tant que tel il existe une relation implicite entre les valeurs d'une rangée. Par exemple, toutes les valeurs dans la première rangée s'appliquent à la première opinion, tandis que les valeurs dans la deuxième rangée s'appliquent à la deuxième opinion et ainsi de suite. Lors de l'ajout d'une nouvelle opinion, toutes les valeurs des données ajoutées doivent apparaître dans la même rangée indépendamment des valeurs vides déjà présentes.

Comme l'outil Importer ne connait pas la relation entre les colonnes, un nouvel

attribut, l'attribut group, a été ajouté pour permettre de lier les colonnes entre elles. L'attribut est également utilisé pour définir la position de la rangée étant la première rangée à laquelle les données peuvent être ajoutées pour toutes les colonnes dans le groupe. Pour les données XML, l'attribut group apparaît dans des éléments `<tuple>` avec l'attribut row. Pour CSV, l'attribut group est placé après l'attribut row. Il est seulement utile d'établir un group lorsque l'attribut row est `row="+"`.

Considérer la mise à jour suivante où un group est spécifié :

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+" group="opinion">
                <atom>append source 1</atom>
            </tuple>
        </table>
        <table name='LatLatLongDetermination_tab'>
            <tuple row="+" group="opinion">
                <atom>append determination 1</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

irn	LatSource_tab(+ group='opinion')	LatLatLongDetermination_tab(+ group='opinion')
5000432	append source 1	append determination 1

Remarquer comment l'attribut group est défini pour les deux colonnes *LatSource_tab* et *LatLatLongDetermination_tab*. La valeur group peut être n'importe quelle chaîne de caractères. Toutes les colonnes avec la même valeur group sont examinées pour déterminer la position de la rangée à utiliser lors de l'ajout des données. Plus d'un group peut être utilisé lorsque des groupes disjoints (indépendants) de colonnes liées existent. Chaque ensemble disjoint de colonnes liées devrait avoir son propre nom de group. La mise à jour ci-dessus produira :

Avant

1	source 1	determination 1
2	source 2	determination 2
3	source 3	

Après

1	source 1	determination 1
2	source 2	determination 2
3	source 3	
4	append source 1	append determination 1

ce qui est probablement ce qui était voulu en premier lieu. Les groupes peuvent

être utilisés avec des tables imbriquées de la même manière qu'ils peuvent l'être avec des tables. Un group peut être appliqué aux tuples extérieurs et/ou intérieurs de la table imbriquée. La mise à jour suivante assure que les valeurs de Source (*LatSource_tab*), Determination (*LatLatLongDetermination_tab*), Latitude (*LatLatitude_nesttab*) et Longitude (*LatLongitude_nesttab*) sont ajoutées à la même rangée :

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+" group="opinion">
                <atom>append source 1</atom>
            </tuple>
        </table>
        <table name='LatLatLongDetermination_tab'>
            <tuple row="+" group="opinion">
                <atom>append determination 1</atom>
            </tuple>
        </table>
        <table name='LatLatitude_nesttab'>
            <tuple row="+" group="opinion">
                <table>
                    <tuple>
                        <atom>new opinion 1, latitude 1</atom>
                    </tuple>
                    <tuple>
                        <atom>new opinion 1, latitude 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
        <table name='LatLongitude_nesttab'>
            <tuple row="+" group="opinion">
                <table>
                    <tuple>
                        <atom>new opinion 1, longitude 1</atom>
                    </tuple>
                    <tuple>
                        <atom>new opinion 1, longitude 2</atom>
                    </tuple>
                </table>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

Le tableau a été tourné sur le côté pour aisance de visualisation.

irn	5000432
LatSource_tab(+ group='opinion')	append source 1
LatLatLongDetermination_tab(+ group='opinion')	append determination 1
LatLatitude_nesttab(+ group='opinion':1)	new opinion 1, latitude 1
LatLatitude_nesttab(+ group='opinion':2)	new opinion 1, latitude 2
LatLongitude_nesttab(+ group='opinion':1)	new opinion 1, longitude 1
LatLongitude_nesttabb(+ group='opinion':2)	new opinion 1, longitude 2

Remarquer comment l'attribut group est appliqué à la rangée extérieure pour s'assurer que les nouvelles valeurs latitude et longitude sont ajoutées à la même rangée que les valeurs Source et Determination :

Avant

1	1	source 1	determination 1	opinion 1, latitude 1	opinion 1, longitude 1
	2			opinion 1, latitude 2	opinion 1, longitude 2
	3			opinion 1, latitude 3	opinion 1, longitude 3
2	1	source 2		opinion 2, latitude 1	opinion 2, longitude 1

Après

1	1	source 1	determination 1	opinion 1, latitude 1	opinion 1, longitude 1
	2			opinion 1, latitude 2	opinion 1, longitude 2
	3			opinion 1, latitude 3	opinion 1, longitude 3
2	1	source 2		opinion 2, latitude 1	opinion 2, longitude 1
3	1	append source 1	append determination 1	new opinion 1, latitude 1	new opinion 1, longitude 1
	2			new opinion 1, latitude 2	new opinion 1, longitude 2

Le point final à voir concernant les groupes est que la position de la rangée à laquelle les nouvelles données seront ajoutées est calculée la première fois que le group est rencontré dans le fichier de données. Une fois que la position de la rangée est déterminée, elle est utilisée pour toutes les colonnes dans le groupe pour l'enregistrement d'importation actuel. Cela signifie que le nom du groupe utilisé pour lier les colonnes ensemble ne devrait apparaître que sur la première rangée (row) à annexer, plutôt que sur chaque rangée. Si le nom du groupe apparaît sur chaque rangée, chaque valeur de mise à jour écrasera la valeur précédente. Considérer la mise à jour :

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+" group="opinion">
                <atom>append source 1</atom>
            </tuple>
            <tuple row="+" group="opinion">
                <atom>append source 2</atom>
            </tuple>
        </table>
        <table name='LatLatLongDetermination_tab'>
            <tuple row="+" group="opinion">
                <atom>append determination 1</atom>
            </tuple>
            <tuple row="+" group="opinion">
                <atom>append determination 2</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

Le tableau a été tourné sur le côté pour aisance de visualisation.

irn	5000432
LatSource_tab(+ group='opinion')	append source 1
LatSource_tab(+ group='opinion')	append source 2
LatLatLongDetermination_tab(+ group='opinion')	append determination 1
LatLatLongDetermination_tab(+ group='opinion')	append determination 2

Cela a pour résultat :

Avant

1	source 1	determination 1

Après

1	source 1	determination 1

2	source 2	

2	source 2	
3	append source 2	append determination 2

Vous avez sans doute remarqué que la valeur `append source 2` a écrasé la valeur `append source 1`. La raison est que quand le groupe `opinion` a été rencontré, il y avait deux rangées dans *LatSource_tab*. Donc le groupe `opinion` fait référence à la troisième rangée. Comme le deuxième `<tuple>` utilise le même nom de groupe que le premier, ils font tous les deux références à la rangée utilisée par le groupe `opinion` (la troisième rangée dans ce cas). Donc la deuxième valeur écrase la première. Si vous souhaitez ajouter plusieurs valeurs, où les colonnes sont liées, un nom de groupe distinct doit être donné pour chaque rangée à annexer. C'est à dire :

XML

```
<table table='ecollectionevents'>
    <tuple>
        <atom name='irn'>5000432</atom>
        <table name='LatSource_tab'>
            <tuple row="+" group="opinion 1">
                <atom>append source 1</atom>
            </tuple>
            <tuple row="+" group="opinion 2">
                <atom>append source 2</atom>
            </tuple>
        </table>
        <table name='LatLatLongDetermination_tab'>
            <tuple row="+" group="opinion 1">
                <atom>append determination 1</atom>
            </tuple>
            <tuple row="+" group="opinion 2">
                <atom>append determination 2</atom>
            </tuple>
        </table>
    </tuple>
</table>
```

CSV

Le tableau a été tourné sur le côté pour aisance de visualisation.

irn	5000432
LatSource_tab(+ group='opinion 1')	append source 1
LatSource_tab(+ group='opinion 2')	append source 2
LatLatLongDetermination_tab(+ group='opinion 1')	append determination 1
LatLatLongDetermination_tab(+ group='opinion 2')	append determination 2

Qui a pour résultat :

Avant

1	source 1	determination 1
2	source 2	

Après

1	source 1	determination 1
2	source 2	
3	append source 1	append determination 1
4	append source 2	append determination 2

Index