



Documentation EMu

Sécurité dynamique

Document Version 1.2

EMu Version 4.0



Contents

SECTION 1	Sécurité dynamique	1
SECTION 2	Security Update	3
	Exemples	5
	Ce qui se passe dans en coulisses	8
	L'ordre de traitement	9
SECTION 3	Column Access Modifier	11
	Exemples	13
SECTION 4	Mandatory Modifier	17
	Exemples	18
SECTION 5	Conclusion	21
	Index	23

SECTION 1

Sécurité dynamique

Le modèle de sécurité par défaut utilisé par EMu est statique dans sa nature. Les privilèges des utilisateurs et des groupes sont définis dans le Registre EMu et chargés dans un module quand il est invoqué. Une fois invoqué la sécurité du module reste la même tout au long de sa durée de vie. Si un utilisateur peut modifier le contenu d'un champ donné, alors il peut le changer pour tous les enregistrements (en supposant que la Sécurité au niveau de l'enregistrement permette que l'enregistrement soit modifié). Dans certains cas, il peut être utile de permettre à certains paramètres de sécurité d'être modifié en fonction des informations stockées dans l'enregistrement en cours.

Par exemple :

1. Le personnel de stockage dans un groupe appelé Stockage est capable de changer tous les champs pour les enregistrements de type `Caisse` et `Cadre`, tandis que pour les enregistrements `Objet`, ils ne peuvent mettre à jour que les informations de dimensions. Le modèle de sécurité actuel d'EMu ne fournit pas un mécanisme pour la mise en œuvre de cette condition avec des paramètres de Registre. Il est possible de « coder en dur » de telles fonctionnalités dans le client, cependant cela devient très difficile à changer lorsque de nouvelles exigences surgissent.
Ce qui serait utile serait d'avoir un mécanisme permettant d'accéder à une colonne à modifier en fonction du contenu de l'enregistrement.
2. De la même façon, vous pouvez avoir besoin que certains champs soient remplis en fonction du type d'enregistrement. Pour les enregistrements `Objet` vous pouvez avoir besoin que le Titre principal soit spécifié, alors que pour les enregistrements `Caisse` et `Cadre`, le titre ne devrait pas être spécifié (en fait, il ne devrait même pas être visible). L'entrée de Registre Mandatory d'EMu permet de définir un champ comme étant obligatoire, cependant il n'est pas possible d'utiliser cette entrée de Registre pour spécifier des paramètres obligatoires conditionnels.
Comme avec le premier exemple, il serait utile de permettre de spécifier le paramètre obligatoire d'un champ en fonction du contenu de l'enregistrement.
3. Vous voulez peut-être modifier les paramètres de la Sécurité au niveau de l'enregistrement en fonction du contenu des données au sein de l'enregistrement lorsque l'enregistrement est sauvegardé. Un exemple peut être que les enregistrements dont le statut de l'enregistrement est mis est à `Retiré` ne peuvent être modifiés que par les utilisateurs dans le groupe Admin. Une telle fonctionnalité peut être « codée en dur » dans le serveur de la base de données.
Cependant, une solution qui utilise le Registre fournirait un mécanisme plus souple.

Avec l'ajout de trois entrées de Registre, EMu 4.1 implémente un modèle de sécurité flexible et dynamique qui peut s'adapter en fonction des données stockées dans un enregistrement :

- L'entrée de Registre Column Access Modifier gère le premier exemple ci-dessus, c'est à dire la possibilité de modifier l'accès des colonnes en fonction du contenu de l'enregistrement.

- L'entrée de Registre Mandatory Modifier gère le deuxième exemple ci-dessus, c'est à dire la possibilité d'ajuster les paramètres obligatoires en fonction du contenu de l'enregistrement.
- L'entrée de Registre Security|Update gère le deuxième exemple, c'est à dire la possibilité de modifier la Sécurité au niveau de l'enregistrement en fonction du contenu de l'enregistrement en cours.

Dans ce document, nous allons voir les trois entrées de Registre et expliquer comment elles peuvent être utilisées pour fournir un modèle de sécurité dynamique, c'est à dire un qui change en fonction des données de l'enregistrement en cours.

SECTION 2

Security Update

L'entrée de registre Security Update permet au contenu d'un ou plusieurs champs d'être modifié en fonction de la valeur dans un champ à chaque fois qu'un enregistrement est sauvegardé (inséré ou modifié). Fait important, les champs de la Sécurité au niveau de l'enregistrement :

- SecCanDisplay
- SecCanEdit
- SecCanDelete

peuvent être ajustés, permettant de modifier la sécurité de l'enregistrement. Mais l'entrée de Registre ne se limite pas à ces champs et n'importe quelle combinaison de champs peut être ajustée.

Le format de l'entrée de Registre est :

User|*user*|Table|*table*|Security|Update|*column*|*value*|*settings*

User|*user*|Table|Default|Security|Update|*column*|*value*|*settings*

Group|*group*|Table|*table*|Security|Update|*column*|*value*|*settings*

Group|*group*|Table|Default|Security|Update|*column*|*value*|*settings*

Group|Default|Table|*table*|Security|Update|*column*|*value*|*settings*

Group|Default|Table|Default|Security|Update|*column*|*value*|*settings*

où :

column définit le champ qui doit être consulté pour trouver une *value* correspondante.

value est un mode de recherche basé sur EMu.

S'il y a une correspondance entre les données dans *column* avec la requête *value*, alors l'entrée de Registre est utilisée et les *settings* sont appliqués.



Si *column* contient un tableau de valeurs, chaque entrée est comparée à *value*.

Puisque *value* est un mode de recherche, une attention particulière est requise si vous voulez une correspondance avec le contenu complet d'un champ.

Par exemple, pour obtenir Collection mais pas Non-collection, il faut utiliser le mode `^Collection$`. Il est important de se rappeler que la *value* opère de la même manière qu'un terme de recherche EMu.

Toutes les comparaisons de *value* sont insensibles à la casse.

settings est une liste d'affectations à des colonnes, séparées par des points-virgules, qui est appliquée s'il y a une correspondance entre les données dans *column* avec la requête *value*.

Le format des *settings* est

column=[+/-] *term*: [+/-] *term*: . . . ; *column*=[+/-] *term*: . . .

où :

column le nom de la colonne à modifier.

term est la valeur à mettre dans *column*.

- Si *term* est précédé par un symbole plus (+), la valeur est ajoutée à la liste des valeurs existantes.
- Si *term* est précédé par un symbole moins (-), la valeur est enlevée de la liste de valeurs existantes.
- Si aucun symbole ne précède un *term*, le contenu actuel de la *column* est supprimé et remplacé par *term*.
- Si plus d'un *term* est fourni pour une colonne donnée, séparés par deux points (:), chaque *term* est appliqué l'un après l'autre.
- Si plus d'une colonne doit être modifiée, chaque ensemble de paramètres de colonne est séparé par un point-virgule (;).

Exemples

Si nous voulons que la Sécurité au niveau de l'enregistrement soit ajustée afin que les utilisateurs dans le groupe Admin soient les seuls autorisés à modifier et supprimer des enregistrements qui ont un Statut de l'enregistrement de *Retired* (*Retiré*), l'entrée suivante peut être utilisée :

Champ	Valeur
Clé 1	Group
Clé 2	Default
Clé 3	Table
Clé 4	Default
Clé 5	Security
Clé 6	Update
Clé 7	SecRecordStatus
Clé 8	^Retired\$
Valeur	SecCanEdit=Group Admin;SecCanDelete=Group Admin

Les clés 7 et 8 indiquent que l'entrée ne s'applique que lorsque le champ *SecRecordStatus* correspond au mode *^Retired\$* (c'est à dire que le champ contient uniquement *Retired*). Si ce n'est pas le cas, l'entrée de Registre est ignorée. Si l'enregistrement correspond aux critères, le champ *SecCanEdit* est réglé sur *Group Admin*. Comme il n'y a pas de plus ni de moins de fourni, le contenu de *SecCanEdit* est remplacé par *Group Admin*. Une mise à jour similaire se produit pour *SecCanDelete*.

Un exemple plus complexe consiste à enlever l'accès d'édition pour tous les utilisateurs Conservation et Storage quand un objet est retiré de l'inventaire (deaccessioned). Supposons que le champ *RecObejctStatus* contienne le mot *Deaccessioned* pour les objets qui ne font plus partie de notre collection. Une entrée de Registre appropriée est :

Champ	Valeur
Clé 1	Group
Clé 2	Default
Clé 3	Table
Clé 4	ecatalogue
Clé 5	Security
Clé 6	Update
Clé 7	RecObjectStatus
Clé 8	^Deaccessioned\$
Valeur	SecCanEdit=-Group Conservation:-Group Storage

Notez que les termes à établir sont précédés d'un signe moins, indiquant que le terme (dans ce cas le nom du groupe) doit être enlevé du champ *SecCanEdit*.

Un troisième exemple nécessite que pour tous les enregistrements avec une estimation de l'objet de *High* (élevé) le groupe *Student* (Étudiant) soit enlevé et le groupe *Valuers* (Experts) soit ajouté pour l'affichage et l'édition de l'enregistrement. Le champ contenant l'estimation de l'objet est *ValValuationCode*. Une entrée de Registre appropriée est :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Security
<i>Clé 6</i>	Update
<i>Clé 7</i>	ValValuationCode
<i>Clé 8</i>	^High\$
<i>Valeur</i>	SecCanDisplay=-Group Student:+Group Valuers; SecCanEdit=-Group Student:+Group Valuers

Remarquez comment plus d'un champ peut être mis à jour avec une seule entrée de Registre. Notez également :

- Si un terme est précédé d'un symbole plus et le terme apparaît déjà dans le champ, il n'est pas rajouté.
- De même, si un terme est précédé d'un moins et qu'il n'apparaît pas dans le champ, il est ignoré.

Dans ce dernier exemple, nous restreignons le privilège d'affichage pour les enregistrements qui n'ont pas été approuvés pour l'affichage sur l'intranet aux groupes *Admin*, *Curator*, *Storage* et *Conservation*. Une fois que l'enregistrement a été approuvé pour l'affichage sur l'intranet, nous allons autoriser tous les utilisateurs à consulter l'enregistrement. Dans ce cas, deux entrées de Registre sont nécessaires :

- La première limite l'accès pour les enregistrements qui ne sont pas disponibles sur l'intranet.
- La deuxième donne accès à tous les utilisateurs pour les enregistrements disponibles sur l'intranet.

Des entrées de Registre appropriées sont :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Security
<i>Clé 6</i>	Update
<i>Clé 7</i>	AdmPublishWebPasswordFlag

Clé 8 N
Valeur ScaCanDisplay=Group Admin:+Group Curator:+Group Storage:+Group Conservation

Champ	Valeur
-------	--------

Clé 1	Group
Clé 2	Default
Clé 3	Table
Clé 4	ecatalogue
Clé 5	Security
Clé 6	Update
Clé 7	AdmPublishWebPasswordFlag
Clé 8	Y
Valeur	ScaCanDisplay=Group Default

Notez que le premier terme de la première entrée de Registre (Group Admin) n'est pas précédé d'un plus ou d'un moins, ce qui signifie qu'il remplace le contenu actuel de *SecCanDisplay*. Les termes suivants doivent être précédés d'un symbole plus sinon ils effaceront également le contenu actuel plutôt que d'ajouter au premier terme. La deuxième entrée du Registre remplace le contenu de *SecCanDisplay* avec group Default (ce qui donne accès à tout le monde).

En utilisant une combinaison d'entrées de Registre, il est possible de produire des modifications de privilèges sophistiquées et dynamiques.

Ce qui se passe en coulisses

L'entrée de Registre Security Update est une entrée de Registre basée sur la Sécurité au niveau de l'enregistrement et, comme toutes les entrées de Registre Sécurité au niveau de l'enregistrement, elle est appliquée par le moteur de base de données. Le fichier `security` stocké dans le répertoire des tables fournit la configuration utilisée pour la Sécurité au niveau de l'enregistrement. Le format XML du fichier `security` a été étendu pour accommoder le contenu de l'entrée de Registre Security Update. Considérez cette entrée :

Champ	Valeur
Clé 1	Group
Clé 2	Default
Clé 3	Table
Clé 4	ecatalogue
Clé 5	Security
Clé 6	Update
Clé 7	SecRecordStatus
Clé 8	^Retired\$
Valeur	SecCanEdit=Group Admin:+Group Registration;SecCanDelete=Group Admin:+Group Registration

Le XML supplémentaire généré dans le fichier `security` pour cette entrée est :

```
<updates>
  <update name="SecRecordStatus" value="Retired">
    <columns>
      <column name="SecCanEdit">
        <values>
          <value operation="replace" term="Group Admin"/>
          <value operation="add" term="Group Registration"/>
        </values>
      </column>
      <column name="SecCanDelete">
        <values>
          <value operation="replace" term="Group Admin"/>
          <value operation="add" term="Group Registration"/>
        </values>
      </column>
    </columns>
  </update>
</updates>
```

Comme vous pouvez le voir le XML suit le séquençage de l'entrée de Registre. Si plusieurs entrées de Registre existent, les balises `<update>` sont répétées.

La bonne nouvelle est que vous n'avez pas besoin d'ajouter le code XML dans le fichier `security`. Quand une entrée de Registre de type `security` est ajoutée ou modifiée dans EMu, le fichier `security` est reconstruit automatiquement, et tout ce que vous avez à faire est d'ajouter les entrées de Registre.

L'ordre de traitement

Le serveur de la base de données applique les paramètres Security Update chaque fois qu'un enregistrement est enregistré (pour toutes les insertions et mises à jour). Les entrées sont appliquées après que les opérations d'affectation soient exécutées et **avant** que la validation soit lancée. Cela signifie que les opérations d'affectation peuvent être utilisées pour construire une valeur composite qui peut être testée par les paramètres Security Update. Par exemple, il est possible de concaténer deux champs en un seul qui peut alors être vérifié.

Il est également possible d'appliquer une formule sophistiquée pour calculer un champ à vérifier. Par exemple, vous voulez peut-être définir des entrées de Registre Security Update basées sur une gamme d'estimations pour un objet. Vous pouvez utiliser une opération d'affectation pour définir une valeur dans un champ basé sur les gammes :

Gamme	Valeur
\$0 - \$1000	Cheap
\$1001 - \$10000	Average
\$10001 -	Pricey

Vous pouvez alors utiliser les trois valeurs, Cheap (pas cher), Average (moyen) et Pricey (cher) dans les entrées de Registre Security Update. Notez que vous ne pouvez pas utiliser le code de validation pour calculer des valeurs puisque les entrées Security Update sont appliquées avant que le code de validation soit exécuté.

SECTION 3

Column Access Modifier

Comme son nom l'indique, l'entrée de Registre Column Access Modifier est utilisée pour modifier les valeurs par défaut de Column Access en fonction des données trouvées dans l'enregistrement en cours.

Le format de l'entrée de Registre est :

User|*user*|Table|*table*|Column Access Modifier|*column*|*value*|*settings*

User|*user*|Table|Default|Column Access Modifier|*column*|*value*|*settings*

Group|*group*|Table|*table*|Column Access Modifier|*column*|*value*|*settings*

Group|*group*|Table|Default|Column Access Modifier|*column*|*value*|*settings*

Group|Default|Table|*table*|Column Access Modifier|*column*|*value*|*settings*

Group|Default|Table|Default|Column Access Modifier|*column*|*value*|*settings*

où :

column définit le champ qui doit être consulté pour trouver une *value* correspondante.

value est une correspondance insensible à la casse (c'est à dire les modes de recherche/caractères génériques ne sont pas autorisés, contrairement à l'entrée Security Update). La *value* est comparée à l'ensemble du contenu de la *column*. La comparaison n'est pas sensible à la casse. Si *column* est un tableau de valeurs, chaque entrée dans le tableau est testée. Si une correspond, l'entrée de Registre s'applique. Les valeurs NULL et NOT NULL peuvent être utilisées pour représenter des valeurs vides et non-vides, respectivement.

settings est une liste d'affectations à des colonnes, séparées par des points-virgules, qui est appliquée s'il y a une correspondance entre les données dans *column* avec la requête *value*.

Le format des *settings* est :

column=[+/-]*perm*: [+/-]*perm*: . . . ; *column*=[+/-]*perm*: . . .

où :

column est le nom de la colonne à modifier.

perm est la permission à ajuster.

- Un signe plus est utilisé pour ajouter une permission.
- Un signe moins enlève une permission.
- Si aucun signe n'est fourni, les permissions actuelles sont remplacées (cela fonctionne de la même façon que pour l'entrée Security Update).

La liste des valeurs possibles pour *perm* est :

- dvDisplay – voir la colonne lors de l'affichage d'un enregistrement.
- dvEdit - voir la colonne lors de la modification d'un

enregistrement.

- `dvInsert` - voir la colonne lors de l'insertion d'un enregistrement.
- `dvQuery` - voir la colonne lors de la recherche.
- `duEdit` - la colonne peut être modifiée pendant la modification d'un enregistrement.
- `duInsert` - la colonne peut être modifiée pendant l'insertion d'un enregistrement.
- `duQuery` - la colonne peut être modifiée lors de la recherche.
- `duReplace` - la colonne peut être utilisée pour un Remplacement Global.

Exemples

Pour cet exemple, nous voulons nous assurer que le champ d'emplacement actuel ne peut pas être mis à jour pour les objets retirés de l'inventaire (deaccessioned), c'est à dire que nous voulons enlever la permission `duEdit` quand le statut de l'objet est `deaccessioned`. Une entrée appropriée est :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access Modifier
<i>Clé 6</i>	RecObjectStatus
<i>Clé 7</i>	Deaccessioned
<i>Valeur</i>	LocCurrentLocation=-duEdit

Désormais quand les objets retirés de l'inventaire sont affichés, le champ d'emplacement actuel est grisé (ce qui indique qu'il ne peut pas être modifié). Pour limiter l'entrée ci-dessus pour les utilisateurs du groupe Curator uniquement, vous aurez besoin de changer la Clé 2 avec `Curator`.

L'entrée de Registre Column Access Modifier est généralement établie par groupes ou utilisateurs.

Contrairement à l'entrée de Registre Security Update, où les changements sont appliqués quand un enregistrement est sauvegardé, les entrées Column Access Modifier sont appliquées immédiatement. Si le contenu d'une colonne est changé et correspond à une entrée, l'entrée est appliquée aussitôt. De plus, après l'application des entrées Column Access Modifier, chaque colonne affectée par des changements précédents, mais non mise à jour avec les changements actuels sera rétablie à sa valeur par défaut (tel que défini par le paramètre Column Access Registry).

Dans l'exemple suivant, nous voulons que le groupe Student (Étudiant) puisse changer le champ *Notes* pour les enregistrements qui ne sont pas retirés de l'inventaire. Pour fournir ce paramètre les paramètres par défaut de Column Access doivent avoir `duEdit` d'activé pour le groupe Student. Les entrées de Registre requises sont :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Student
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access
<i>Clé 6</i>	NotNotes

Valeur dvQuery;dvDisplay;dvEdit;dvInert;duEdit;duInsert;duQuery;duReplace

Champ	Valeur
-------	--------

Clé 1 Group

Clé 2 Student

Clé 3 Table

Clé 4 ecatalogue

Clé 5 Column Access Modifier

Clé 6 RecObjectStatus

Clé 7 Deaccessioned

Valeur NotNotes=-duEdit

La première entrée définit les permissions par défaut pour la colonne *NotNotes*. Notez que `duEdit` est activé par défaut. Cela permet aux utilisateurs de groupe Student de modifier le contenu du champ.

La deuxième entrée modifie les paramètres par défaut pour désactiver `duEdit` quand l'objet est retiré de l'inventaire.

Les entrées ci-dessus montrent comment l'entrée de Registre Column Access Modifier peut être utilisée avec l'entrée de Registre Column Access pour fournir un ensemble prévisible de permissions pour toutes les valeurs du statut de l'objet.

Dans ce dernier exemple, nous désactivons la modification du champ *Other Titles* (*Autres Titres*) pour tous les utilisateurs dans le groupe Curator jusqu'à ce qu'une valeur soit entrée dans le champ *Title*. La restriction doit s'appliquer lors de la création d'enregistrements ainsi que lors de leur modification. L'entrée suivante peut être utilisée :

Champ	Valeur
-------	--------

Clé 1 Group

Clé 2 Curator

Clé 3 Table

Clé 4 ecatalogue

Clé 5 Column Access Modifier

Clé 6 RecMainTitle

Clé 7 NULL

Valeur RecOtherTitles=-duEdit:-duInsert

Vous avez peut-être été tentés d'utiliser les entrées suivantes :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Curator
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access Modifier
<i>Clé 6</i>	RecMainTitle
<i>Clé 7</i>	NULL
<i>Valeur</i>	RecOtherTitles=-duEdit:-duInsert

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Curator
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access Modifier
<i>Clé 6</i>	RecMainTitle
<i>Clé 7</i>	NOT NULL
<i>Valeur</i>	RecOtherTitles=+duEdit:+duInsert

pour vous assurer que les utilisateurs puissent modifier le champ d'autres titres, si le titre principal est rempli. La deuxième entrée de Registre force l'activation des privilèges de modification et d'insertion même si les paramètres par défaut de Column Access ne les a pas. C'était peut-être votre intention d'avoir ce comportement, cependant les instructions originales stipulaient uniquement que les permissions de modification et d'insertion devaient être désactivées pour les autres titres si le titre principal est vide. Il n'est pas exprimé ce que les permissions devraient être si le titre principal est rempli. Une autre façon d'écrire la deuxième série d'entrées de Registre est :

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Curator
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access
<i>Clé 6</i>	RecOtherTitles
<i>Valeur</i>	dvQuery;dvDisplay;dvEdit;dvInert;duEdit;duInsert;duQuery;duReplace

Champ	Valeur
<i>Clé 1</i>	Group
<i>Clé 2</i>	Curator
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Column Access Modifier
<i>Clé 6</i>	RecMainTitle
<i>Clé 7</i>	NULL
<i>Valeur</i>	RecOtherTitles=-duEdit:-duInsert

Dans ce cas, les permissions de colonne par défaut activent les privilèges de modification et d'insertion, tandis que la deuxième entrée de Registre les désactive si le titre principal est vide. Si le titre principal est rempli, les permissions par défaut sont appliquées, ce qui permet d'activer les privilèges de modification et d'insertion.

SECTION 4

Mandatory Modifier

L'entrée de Registre Mandatory Modifier est utilisée pour modifier le paramètre obligatoire pour un champ donné en fonction des données contenues dans l'enregistrement en cours.

Le format de l'entrée de Registre est :

User|*user*|Table|*table*|Mandatory Modifier|*column*|*value*|*settings*

User|*user*|Table|Default|Mandatory Modifier|*column*|*value*|*settings*

Group|*group*|Table|*table*|Mandatory Modifier|*column*|*value*|*settings*

Group|*group*|Table|Default|Mandatory Modifier|*column*|*value*|*settings*

Group|Default|Table|*table*|Mandatory Modifier|*column*|*value*|*settings*

Group|Default|Table|Default|Mandatory Modifier|*column*|*value*|*settings*

où :

column définit quel champ doit être consulté pour rechercher une *value* correspondante.

value est une correspondance insensible à la casse (c'est à dire les modes de recherche/caractères génériques ne sont pas autorisés, contrairement à l'entrée Security|Update). La *value* est comparée à l'ensemble du contenu de la *column*. La comparaison n'est pas sensible à la casse. Si *column* est un tableau de valeurs, chaque entrée dans le tableau est testée. Si une correspond, l'entrée de Registre s'applique. Les valeurs NULL et NOT NULL peuvent être utilisées pour représenter des valeurs vides et non-vides, respectivement.

settings est une liste d'affectations à des colonnes, séparées par des points-virgules, qui est appliquée s'il y a une correspondance entre les données dans *column* avec la requête *value*.

Le format des *settings* est :

column=*setting*; *column*= *setting*; . . .

où :

column est le nom de la colonne à modifier.

setting est *true* (la colonne doit être obligatoire) ou *false* (la colonne ne doit pas être obligatoire, c'est à dire supprimer le paramètre obligatoire).

Exemples

Exemple 1

Pour cet exemple, nous voulons que le champ *Titre principal* soit obligatoire, mais seulement si le type d'enregistrement est `Objet`. Une entrée appropriée est :

Field	Value
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Mandatory Modifier
<i>Clé 6</i>	RecObjectType
<i>Clé 7</i>	Objet
<i>Valeur</i>	TitMainTitle=true

Maintenant lorsque le type d'enregistrement est `Objet`, le champ *Titre principal* doit être rempli avant que l'enregistrement puisse être sauvegardé. Si le *Titre principal* n'est pas complété, le message d'erreur standard s'affiche. Vous pouvez personnaliser le message d'erreur affiché en utilisant l'entrée de *Registre Mandatory*. Par exemple :

Field	Value
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Mandatory
<i>Clé 6</i>	TitMainTitle
<i>Valeur</i>	False; Veuillez saisir un Titre principal pour l'objet

affichera le message d'erreur *Veillez saisir un Titre principal pour l'objet* si le champ n'est pas renseigné alors que le paramètre obligatoire (mandatory) est `true`. Une valeur `false` indique que le champ n'est pas obligatoire par défaut.

Exemple 2

L'exemple ci-dessus montre comment modifier le paramètre obligatoire pour un champ (*TitMainTitle*) en fonction de la valeur dans un autre champ (*RecObjectType*). Dans l'exemple suivant, le paramètre obligatoire pour plus d'un champ est modifié en fonction des valeurs de plusieurs champs. Si le type d'enregistrement est *Objet* et le statut de l'objet est *Accessioned*, alors les champs *Numéro d'acquisition*, *Date d'acquisition* et *Lot d'acquisition* doivent être renseignés :

Field	Value
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Mandatory Modifier
<i>Clé 6</i>	RecObjectType
<i>Clé 7</i>	Objet
<i>Valeur</i>	TitAccessionNo=true;TitAccessionDate=true;TitAccessionLot=true

Field	Value
<i>Field</i>	Value
<i>Clé 1</i>	Group
<i>Clé 2</i>	Default
<i>Clé 3</i>	Table
<i>Clé 4</i>	ecatalogue
<i>Clé 5</i>	Mandatory Modifier
<i>Clé 6</i>	TitObjectStatus
<i>Clé 7</i>	Accessioned
<i>Valeur</i>	TitAccessionNo=true;TitAccessionDate=true;TitAccessionLot=true

Lorsque plusieurs valeurs doivent être vérifiées, une entrée de Registre Mandatory Modifier est nécessaire pour chaque valeur pour chaque colonne. Lorsque plusieurs entrées de Registre correspondent aux valeurs dans les champs (comme dans l'exemple précédent où à la fois le type d'objet et le statut de l'objet doivent correspondre) une opération booléenne ET est appliquée aux paramétrages obligatoires. Cela signifie qu'à moins que tous les paramètres pour un champ donné soient *true*, le paramétrage obligatoire (mandatory) est *false*.

SECTION 5

Conclusion

EMu 4.1 voit l'ajout de trois nouvelles entrées de Registre. La première, Security|Update, permet aux permissions au niveau de l'enregistrement d'être modifiées quand un enregistrement est sauvegardé, en fonction du contenu de l'enregistrement. La deuxième, Column Access Modifier, permet aux privilèges des champs d'être modifiés en fonction du contenu de l'enregistrement. La troisième, Mandatory Modifier, permet de spécifier les champs obligatoires en fonction du contenu de l'enregistrement. La combinaison des deux fonctions fournit un mécanisme utile pour modifier les paramètres de sécurité d'EMu dynamiquement. L'utilisation de sécurité dynamique permet de mettre en œuvre des modèles de sécurité très flexibles.

Index

C

Column Access Modifier • 11

Conclusion • 17

D

Dynamic Security • 1

E

Examples • 5, 12

S

Security Update • 3

T

The order of processing • 9

W

What's happening behind the scenes • 8