TIBCO EBX® Documentation
Produit

Version 5.9.22
janvier 2023

TIBCO®

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO and TIBCO EBX are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. Cloud Software Group, Inc. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (*https://www.tibco.com/patents*) for details.

Copyright 2006-2023. Cloud Software Group, Inc. All rights reserved.

Table des matières

Guide utilisateur

Introduction

1. Notions cls de TIBCO EBX 13
2. Interface utilisateur 17
3. Glossaire 23

Modèles de données

4. Introduction aux modles de donnes 34

Implémentation des modèles de données

5. Cration du modle de donnes 39
6. Configuration du modle de donnes 41
7. Modlisation de la structure des donnes 49
8. Proprits des lments du modle de donnes 55
9. Contrles sur les lments du modle de donnes 71
10. Barres d'outils 79
11. Actions sur les modles de donnes existants 87

Publication et gestion de versions des modèles de données

12. Publication du modle de donnes 91
13. Gestion des versions de modles de donnes embarqus 93

Espaces de données

14. Introduction aux espaces de donnes 98
15. Cration d'un espace de donnes 101
16. Actions sur les espaces de donnes existants 103
17. Images 113

Jeux de données

18. Introduction aux jeux de donnes 120
19. Cration du jeu de donnes 123
20. Visualisation des donnes 125
21. Edition des donnes 135
22. Actions sur les jeux de donnes existants 139
23. Hritage entre jeux de donnes 143

Modèles de workflow

24. Introduction aux modles de workflow 148
25. Modlisation du workflow 153
26. Configuration du modle de workflow 165
27. Publication d'un modle de workflow 173

Workflows de données

28. Introduction aux workflows de donnes 176
29. Utilisation de l'interface utilisateur de la section Workflow de donnes 179
30. Bons de travail 185

Gestion de workflows de données
31. Lancement et monitoring de workflows de donnes....191
32. Administration de workflows de donnes....193

Services de données

33. Introduction aux services de donnes....198
34. Gnration de WSDL pour services de donnes....201

Manuel de référence

Intégration

35. Prsentation de l'intgration et des dploiements 207
36. Utiliser TIBCO EBX comme composant web 211
37. Services UI prdfinis 221

Services d'import et d'export

38. Import et export XML 235
39. Import et export CSV 241

40. Syntaxe XPath supporte 249

Localisation

41. Libells et localisation 258
42. Extension de l'internationalisation de TIBCO EBX 261

Persistance

43. Prsentation de la persistance 264
44. Mode relationnel 267
45. Historique 273
46. Rplication 281
47. volutions du modle de donnes 287

Divers

48. Hritage et rsolution de valeur 294
49. Permissions 299
50. diteur de critres 317
51. Instructions relatives aux performances 319

Guide d'administration (en anglais)

52. Administration overview....330

Installation & configuration

53. Supported environments....334
54. Java EE deployment....341

Installation notes

55. Installation note for JBoss EAP 7.1.x....351
56. Installation note for Tomcat 8.x....355
57. Installation note for WebSphere AS 9....359
58. Installation note for WebLogic 12c R2....365

59. TIBCO EBX main configuration file....369
60. Initialization and first-launch assistant....391
61. Deploying and registering TIBCO EBX add-ons....393

Technical administration

62. Repository administration....396
63. UI administration....407
64. Users and roles directory....423
65. Data model administration....427
66. Database mapping administration....429
67. Workflow management....433
68. Task scheduler....437
69. Audit trail....443
70. Other....445

Distributed Data Delivery (D3)

71. Introduction to D3....448
72. D3 broadcasts and delivery dataspaces....453
73. D3 JMS Configuration....457
74. D3 administration....465

Guide de sécurité (en anglais)

75. Security Best Practices.....476

Guide du développeur (en anglais)

Introduction

76. Packaging TIBCO EBX modules....483
77. Mapping to Java....489
78. Tools for Java developers....495
79. Terminology changes....497

Data model

80. Introduction....500
81. Data types....503
82. Tables and relationships....517
83. Constraints, triggers and functions....537
84. Labels and messages....555
85. Additional properties....561
86. Data services....569
87. Toolbars....571
88. Workflow model....573

User interface

89. Interface customization....584

User services

90. Overview....587
91. Quick start....591
92. Implementing a user service....595
93. Declaring a user service....609
94. Development recommendations....615

SOAP data services

95. Introduction....620
96. WSDL generation....631
97. SOAP operations....639

REST data services

98. Introduction....674
99. Built-in RESTful services....681
100. JSON format....719
101. REST Toolkit....745

Guide utilisateur

Introduction

CHAPITRE 1

Notions cls de TIBCO EBX

Ce chapitre contient les sections suivantes :

1. Concepts et outils associs
2. Architecture

1.1 Concepts et outils associs

Le Master Data Management (MDM) est un moyen de modliser, grer et gouverner les donnes partages. Quand des donnes sont partages par plusieurs systmes informatiques, ainsi que des quipes professionnelles diffrentes, l'existence d'une seule version gouverne des donnes de rfrence est essentielle.

Avec EBX, les utilisateurs mtier et les quipes informatiques peuvent collaborer sur une solution unifie, afin de concevoir des modles de donnes et de grer le contenu des donnes de rfrence.

EBX est un logiciel de gestion des donnes de rfrence qui permet de modliser tout type de donnes de rfrence et d'appliquer une gouvernance grce des outils avancs tels que le workflow collaboratif, le contrle de l'dition de donnes, la gestion hirarchique des donnes, le contrle de version, et la scurit.

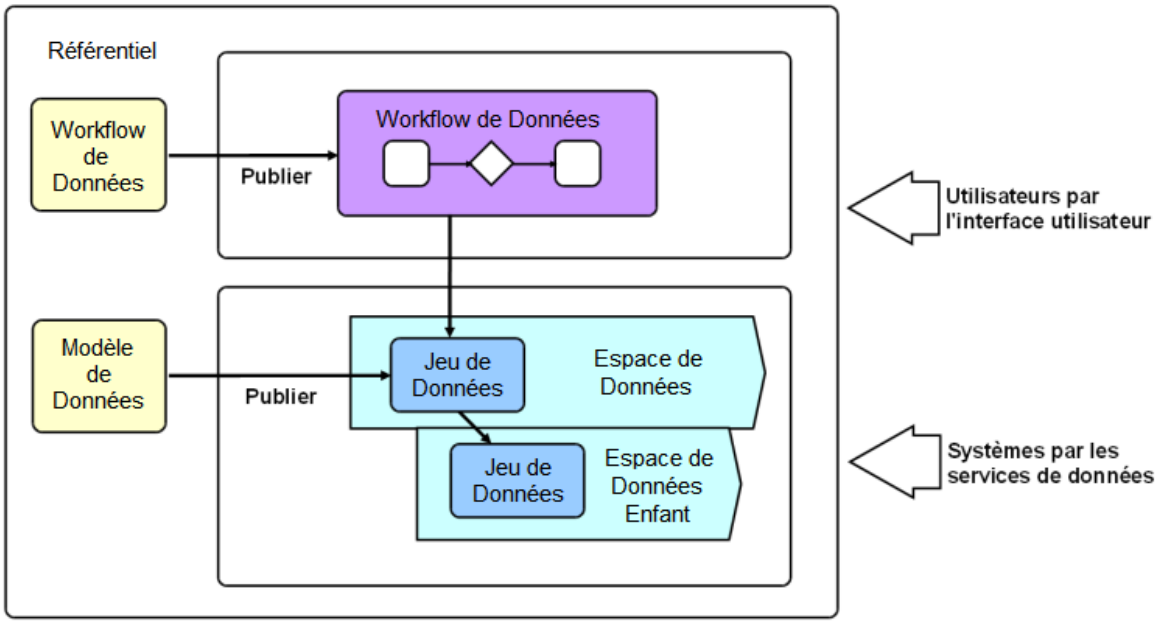
Un projet MDM bas sur EBX commence par la cration d'un *modle de donnes*. Celui-ci dfinit les tables, les champs, les liens et les rgles mtiers permettant de dcrire les donnes de rfrence. De bons exemples sont les catalogues de produits, les hirarchies financires, les listes de fournisseurs ou simplement les tables de rfrence.

Ce modle de donnes peut ensuite tre publi en tant que *jeu de donnes*, stockant le contenu des donnes de rfrence. Les jeux de donnes sont organiss dans des *espaces de donnes*. Un espace de donnes est un conteneur qui permet d'isoler toutes les mises jour effectues. Cela permet de travailler sur plusieurs versions parallles des donnes.

Les *workflows* sont indispensables aux processus de modification et d'approbation sur les donnes. Un workflow permet de modliser un processus tape par tape, comprenant la participation de plusieurs utilisateurs humains et automatiss.

Les *modles de workflow* dfinissent les tches effectuer, ainsi que les participants associs chaque tche. Ds qu'un modle de workflow est publi, il peut tre excut en tant que *workflow de donnes*. Les workflows de donnes permettent d'envoyer aux utilisateurs des notifications concernant les vnements pertinents et les tches accomplir, le tout dans un contexte collaboratif.

Les *services de donnes* aident intgrer EBX des systmes tiers ("middleware"), en leur permettant d'accder aux donnes, ou de grer des espaces de donnes et des workflows.



Voir aussi

Modle de donnes *[p 24]*

Jeu de donnes *[p 26]*

Espace de donnes *[p 28]*

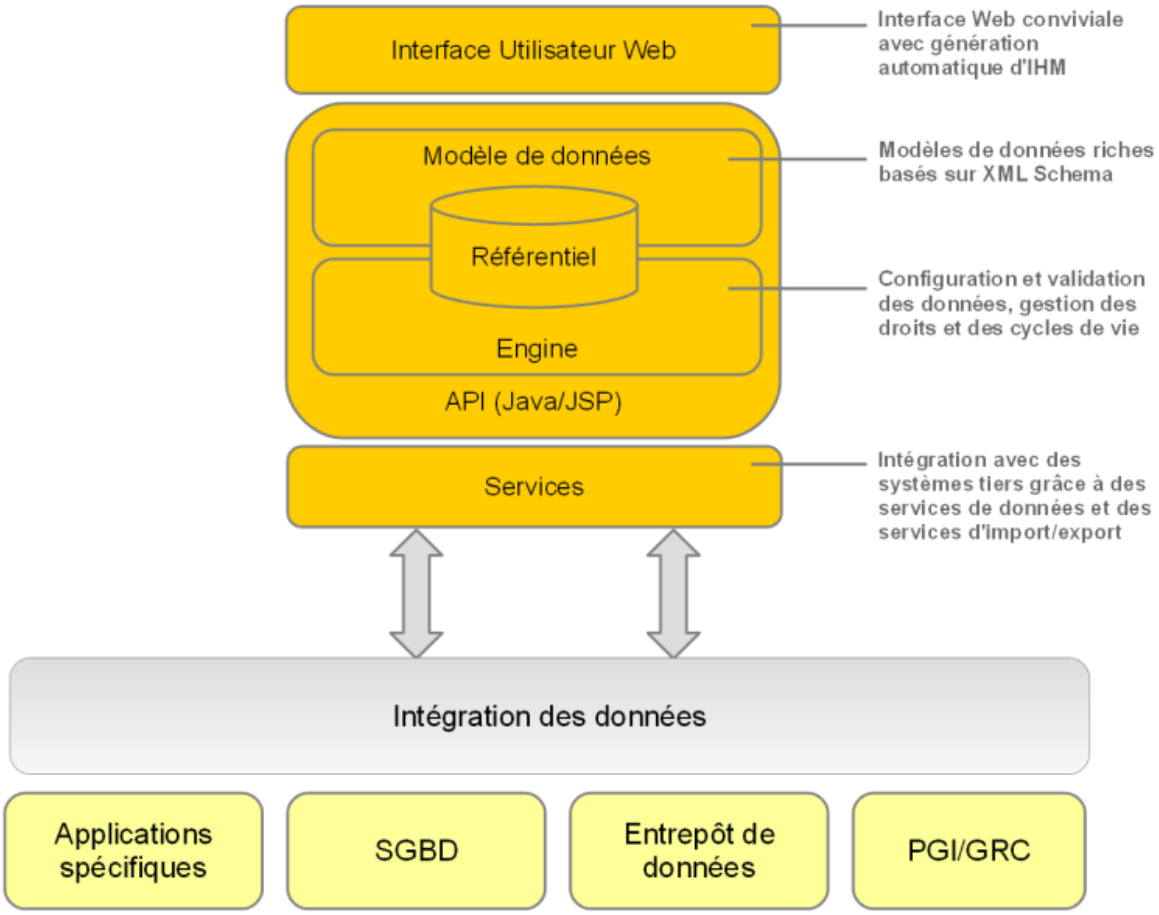
Modle de workflow *[p 29]*

Workflow de donnes *[p 30]*

Service de donnes *[p 31]*

1.2 **Architecture**

Le schma suivant prsente l'architecture de EBX.



CHAPITRE 2

Interface utilisateur

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Perspective avance
3. Perspectives
4. Panneau utilisateur
5. Fonctionnalits de l'interface utilisateur
6. O trouver de l'aide sur EBX

2.1 Prsentation

La prsentation gnrale des espaces de travail sur TIBCO EBX peut tre entirement personnalise par un administrateur.

Lorsque plusieurs perspectives personnalises ont t cres, elles peuvent tre slectionnes via l'icne 'Perspective' dans l'en-tte de l'cran.

La perspective avance est accessible par dfaut.

Voir aussi*UI administration* *[p 407]*

2.2 Perspective avance

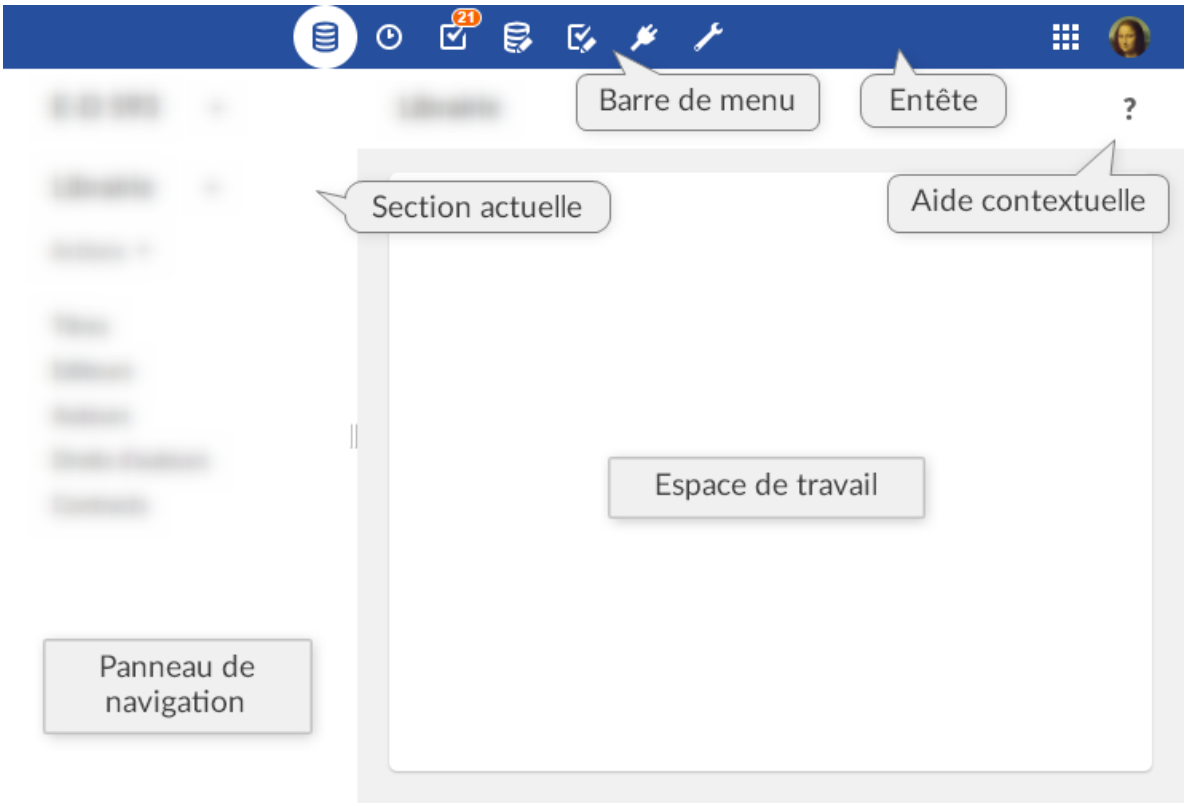
Par dfaut, la perspective avance d'EBX est accessible tous les utilisateurs. Cependant, son accs peut tre restreint certains profils uniquement. Cette vue est divise en plusieurs zones principales, rfrences dans la documentation sous les termes suivants:

> **Note**
>
> La perspective avance est toujours accessible aux utilisateurs via une slection explicite (via un composant Web par exemple). Contrairement aux autres perspectives, elle peut seulement tre « cache » dans l'interface afin que l'utilisateur ne puisse pas l'appliquer lui-mme.

- **En-tte**:l'avatar de l'utilisateur actuel s'affiche dans cette zone, ainsi que l'icne de slection des perspectives. En cliquant sur l'avatar de l'utilisateur, le panneau utilisateur s'ouvre.

- **Barre de menu**:cette zone comprend toutes les fonctionnalits accessibles l'utilisateur actuel et lui permet de naviguer entre elles.
- **Panneau de navigation**:cette zone rsume visuellement les diverses possibilits de navigation. Par exemple : slectionner une table dans un jeu de donnes, ou un bon de travail dans un workflow.
- **Espace de travail**:zone de travail principale dpendant du contexte. Par exemple, la table slectionne dans le panneau de navigation s'affiche dans l'espace de travail, ou bien un bon de travail en cours s'y excute.

Les sections fonctionnelles suivantes sont affiches dans l'interface selon les permissions de l'utilisateur actuel : *Donnes*, *Espace de donnes*, *Modlisation*, *Workflow de donnes*, *Services de donnes*, et *Administration*.



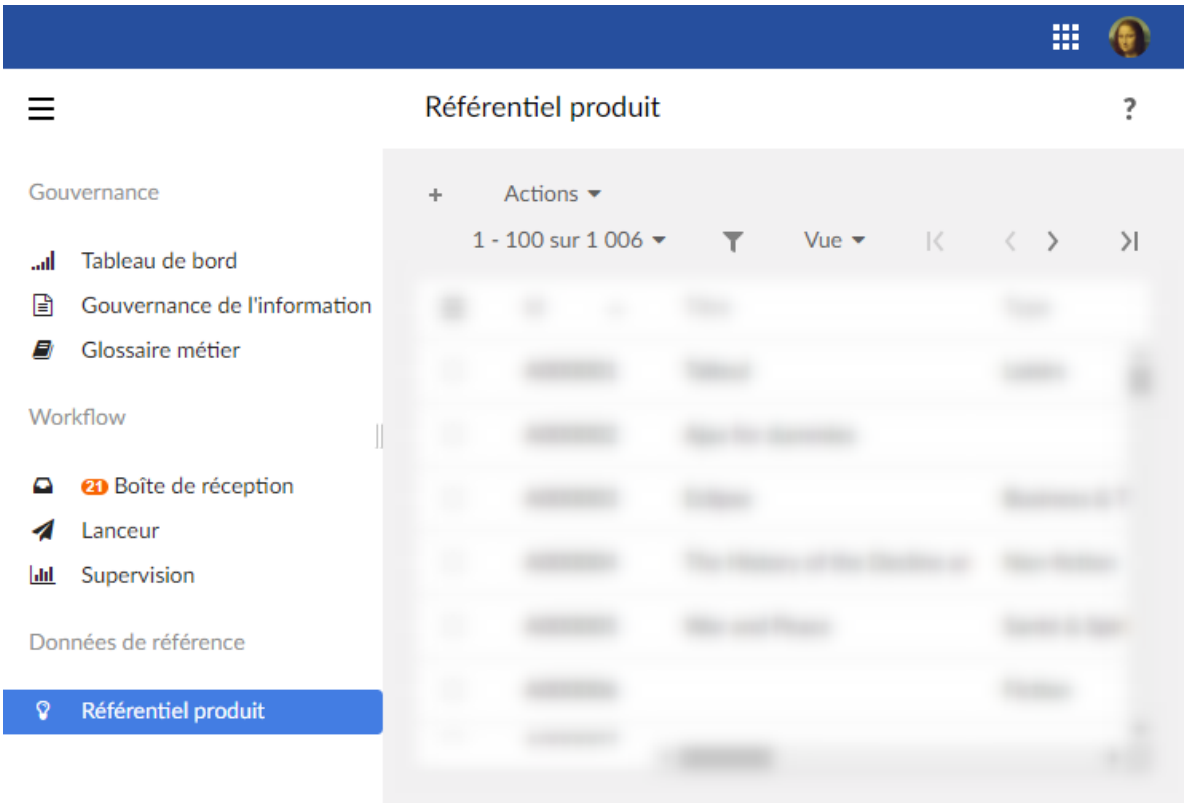
2.3 **Perspectives**

Les perspectives dans EBX sont des vues configurables avec une audience dfinie. Les perspectives permettent aux utilisateurs mtier de bnficier d'une interface simplifie. Une perspective peut tre affecte un ou plusieurs profils. Cette vue est divise en plusieurs zones principales, rfrences dans la documentation sous les termes suivants :

- **En-tte** : l'avatar de l'utilisateur actuel s'affiche dans cette zone, ainsi que l'icne de slection des perspectives (lorsque plusieurs sont disponibles). En cliquant sur l'avatar de l'utilisateur, le panneau utilisateur s'ouvre.
- **Panneau de navigation** : cette zone affiche le menu hirarchique tel qu'il a t configur par l'administrateur de perspectives. Ce panneau peut tre dvelopp ou rduit et permet d'accder aux entits et services correspondant l'activit de l'utilisateur.
- **Espace de travail** : zone de travail principale dpendant du contexte.

Une perspective est configure par un utilisateur ayant les autorisations ncessaires. Pour plus d'informations sur la configuration d'une perspective, voir perspective administration (en anglais) [p 408].

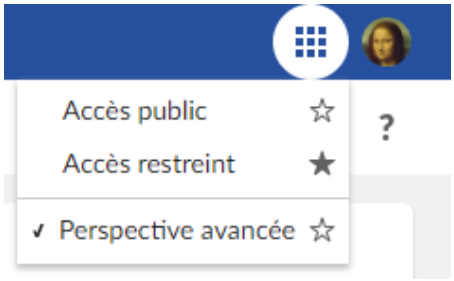
Exemple de menu hirarchique :



Perspectives favorites

Un utilisateur, lorsque plusieurs perspectives ont t dfinies pour son profil, peut dfinir une perspective favorite afin que cette dernire soit slectionne par dfaut sa connexion. Pour ce faire, une icne est prsente ct de chaque perspective dans le slecteur de perspectives:

- Une toile pleine indique la perspective favorite. Un clic sur cette icne dslectionne la perspective favorite.
- Une toile vide indique que la perspective associe n'est pas la favorite. Un clic sur cette icne va dfinir cette perspective comme favorite.



Voir aussi*Recommended perspectives [p 419]*

2.4 **Panneau utilisateur**

Les fonctionnalits gnrales d'EBX sont regroupes dans le panneau utilisateur. Pour y accder, cliquer sur l'avatar (ou les initiales) de l'utilisateur actuel, dans l'en-tte de chaque page.

Le panneau utilisateur affiche alors l'avatar de l'utilisateur et donne accs la configuration du profil (selon les droits de l'utilisateur), la slection de la langue et de la densit d'affichage;et la documentation en ligne.

Attention

Le bouton de dconnexion est situ sur le panneau utilisateur.

Avatar

Un avatar peut tre dfini pour chaque utilisateur. L'avatar est constitu d'une image, dfinie via une URL;ou de deux lettres (par dfaut les initiales de l'utilisateur). La couleur de fond est attribue automatiquement et ne peut pas tre modifie. L'image utilise doit imprativement tre au format carr mais n'est pas limite en termes de taille.

> **Note**
>
> Les avatars s'affichent au niveau du panneau utilisateur, de l'historique et des interfaces du workflow.

La fonctionnalit est galement disponible via la mthode Java `UIComponentWriter.addUserAvatar`[API].

L'affichage de l'avatar peut tre personnalis partir de la section 'Ergonomie et disposition' dans 'Administration'. Il est possible d'afficher l'avatar, le nom de l'utilisateur, ou les deux la fois.

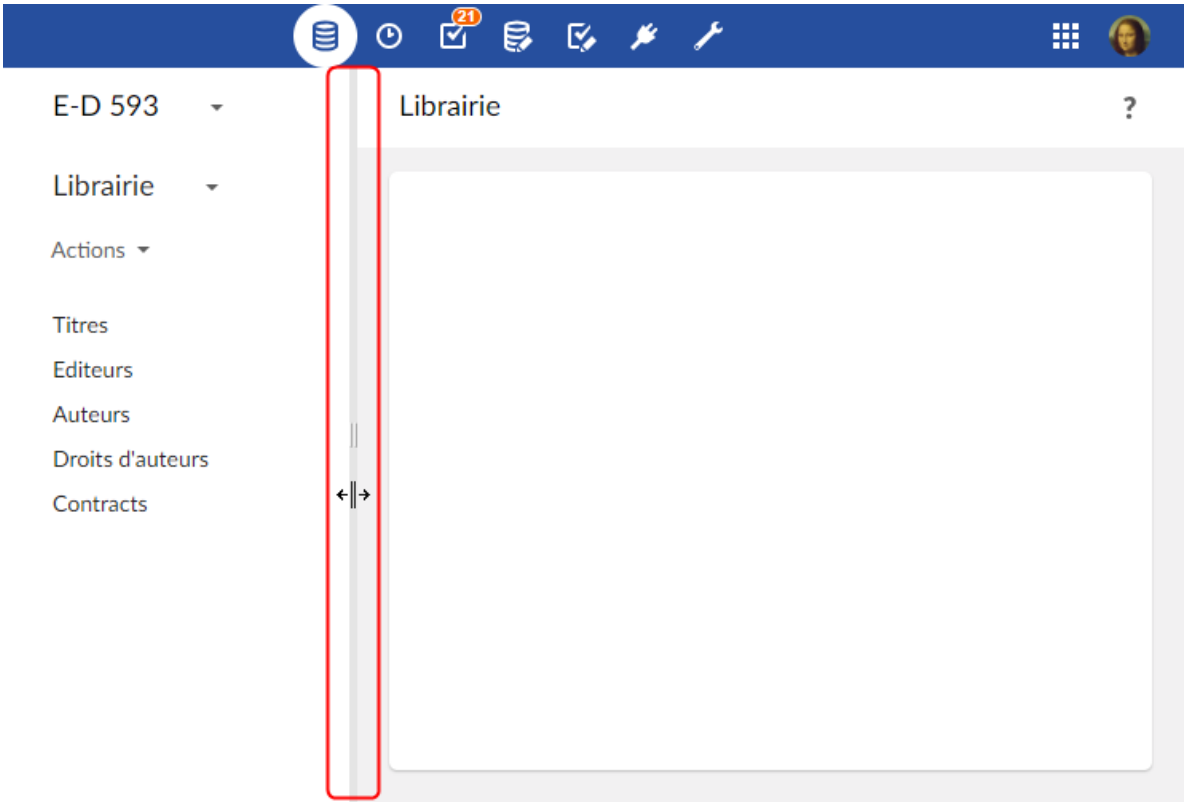
Densit

La densit d'affichage de la police peut tre personnalise : 'Compacte' ou 'Confortable'. Le mode d'affichage peut tre modifi via le panneau utilisateur.

2.5 Fonctionnalits de l'interface utilisateur

Rinitialiser la largeur du panneau de navigation

Si la largeur du panneau de navigation a t modifie, elle peut tre rinitialise en double-cliquant sur la bordure.



2.6 O trouver de l'aide sur EBX

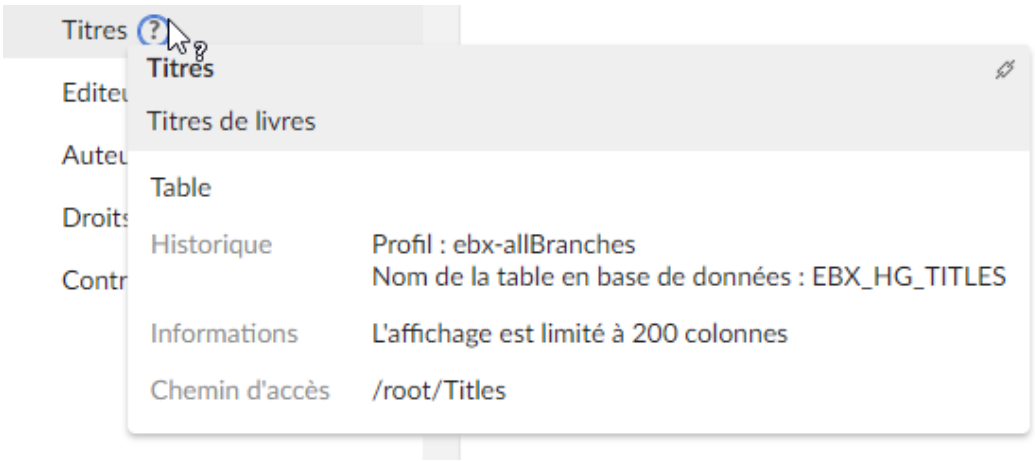
En plus de la documentation complte du produit accessible via le panneau utilisateur [p 20], l'aide est accessible de plusieurs faons dans l'interface.

Aide contextuelle

Dans n'importe quel espace de travail d'EBX, une aide spcifique au contexte actuel est disponible en cliquant sur le point d'interrogation situ droite du second bandeau. Le chapitre correspondant de la documentation produit sera alors affich.

Aide contextuelle sur lments

Lorsque la souris survole un lment pour lequel une aide spcifique a t dfinie, un point d'interrogation apparaît. En cliquant sur le point d'interrogation, un panneau de documentation affiche les informations associes.



Un bouton dans le coin suprieur droit du panneau permet de rcuprer un permalien vers l'lment. Ce bouton n'est pas disponible pour tous les types d'lments.

CHAPITRE 3

Glossaire

Ce chapitre contient les sections suivantes :

1. Gouvernance
2. Modlisation de donnes
3. Gestion des donnes
4. Cycle de vie des donnes
5. Historique
6. Modlisation de workflow
7. Workflows de donnes
8. Services de donnes
9. Transverse

3.1 Gouvernance

rfrentiel

Entit de stockage ct serveur contenant toutes les donnes gres par TIBCO EBX. Le rfrentiel est organis en espaces de donnes.

Voir espace de donnes [p 28].

profil

Terme gnrique dsignant soit un utilisateur, soit un rle. Les profils sont utiliss pour dfinir des rgles de permission et des workflows de donnes.

Voir utilisateur [p 23], rle [p 24].

API Java : `Profile`[API].

utilisateur

Entit cre dans le rfrentiel afin de permettre des personnes physiques ou des systmes externes de s'authentifier et d'accder EBX. Un utilisateur peut avoir un ou plusieurs rles et possder diverses informations de compte telles que nom, prnom, login, e-mail, etc.

Voir annuaire des utilisateurs et des rles [p 24], profil [p 23].

Concept apparent : User and roles directory [p 423].

API Java : `UserReference`[API].

rle

Classification d'utilisateur utilise pour les rgles de permission et les workflows de donnes. Chaque utilisateur peut appartenir plusieurs rles.

Ds qu'un profil de type rle est configur dans EBX, le comportement rsultant de cette configuration s'applique tous les utilisateurs membres de ce rle. Par exemple, dans un modle de workflow, un rle peut tre configur pour le(s) destinataire(s) d'un bon de travail.

Voir annuaire des utilisateurs et des rles [p 24], profil [p 23].

Concept apparent : User and roles directory [p 423].

API Java : `Role`[API].

administrateur

Rle prdfini permettant d'accder l'administration technique et la configuration de EBX.

annuaire des utilisateurs et des rles

Annuaire dfinissant les mthodes disponibles pour l'authentification d'accs au rfrentiel, ainsi que les rles disponibles et les utilisateurs autoriss accder au rfrentiel, avec leurs rles respectifs.

Voir utilisateur [p 23], rle [p 24].

Concept apparent : User and roles directory [p 423].

API Java : `Directory`[API], `DirectoryHandler`[API].

session utilisateur

Contexte d'accs au rfrentiel associ un utilisateur (utilisateur ayant t authentifi par rapport l'annuaire des utilisateurs et des rles).

Concept apparent : User and roles directory [p 423].

API Java : `Session`[API].

3.2 Modlisation de donnes

Section de la documentation Modles de donnes [p 34]

modle de donnes

Dfinition structure des donnes grer dans le rfrentiel EBX. Un modle de donnes dcrit la structure des donnes en termes d'organisation, de type et de relations smantiques. Le but d'un modle de donnes est de dfinir la structure et les caractristiques d'un jeu de donnes, qui est une instance d'un modle de donnes contenant les donnes gres par le rfrentiel.

Voir jeu de donnes [p 26].

Concept apparent : Modles de donnes [p 34].

champ

Elment de base du modle de donnes dfini par un nom et un type de donnes simple. Un champ peut tre directement dfini la racine du modle de donnes, ou en tant que colonne d'une table. Il est possible d'assigner des contraintes de base sur la valeur du champ, par exemple sur sa longueur, ainsi que des rgles de validation plus complexes impliquant des calculs. La valeur du champ peut tre automatiquement calcule l'aide du mcanisme d'hritage de donnes, ou de rgles de calcul. Un champ peut tre dfini comme tant une liste agrge en spcifiant une cardinalit maximale suprieure 1. Chaque lment de la liste ainsi dfinie sera du mme type que le champ initial. Les champs peuvent tre regroups pour faciliter l'organisation du modle de donnes.

Par dfaut, les champs sont reprsents par l'icne .

Voir enregistrement [p 26], groupe [p 25], table (modle de donnes) [p 25], rgle de validation [p 26], hritage [p 27].

Concept apparent : Proprits des lments de structure [p 55], Contrles sur les champs de donnes [p 71].

API Java : `SchemaNode`[API].

L'ancien nom (avant la version 5) de "champ" tait "attribute".

cl primaire

Champ ou composition de plusieurs champs identifiant de manire unique un enregistrement dans une table.

Les cls primaires sont reprsentes par l'icne .

Concept apparent : Tables [p 517].

cl trangre

Champ ou composition de plusieurs champs rfrenant un enregistrement d'une autre table, via sa cl primaire.

Les cls trangres sont reprsentes par l'icne .

Voir cl primaire [p 25].

Concept apparent : Cl trangre [p 522].

table (modle de donnes)

Elment du modle de donnes compos de champs et/ou de groupes. Une table doit au moins tre compose d'un champ dfini comme tant une cl primaire. Une table peut tre utilise pour la cration d'un type rutilisable, afin de crer d'autres lments bass sur la structure de cette table.

Les tables sont reprsentes par l'icne .

Voir enregistrement [p 26], cl primaire [p 25], type rutilisable [p 26].

groupe

Entit de classification utilise pour organiser les donnes du modle. Un groupe peut contenir des champs, d'autres groupes, et des tables. Si un groupe contient des tables, alors celui-ci ne pourra pas tre inclus

dans une autre table. Un groupe peut tre utilis pour la cration d'un type rutilisable afin de crer d'autres lments bass sur la structure de ce groupe.

Les groupes sont reprsents par l'icne .

Voir type rutilisable [p 26].

API Java : `SchemaNode`[API].

type rutilisable

Dfinition d'un type simple ou complexe qui peut tre partage entre diffrents lments d'un modle.

rgle de validation

Association d'une ou plusieurs rgles de contrle dfinies sur un champ ou une table. Toute donne saisie ne respectant pas ces contrles sera dclare invalide, selon la svrit associe la rgle de validation.

L'ancien nom (avant la version 5) de "rgle de validation" tait "constraint".

assistant de modlisation de donnes (DMA)

L'interface utilisateur inclut un outil d'aide la modlisation des donnes. Cet outil permet de dfinir la structure d'un modle, de crer et diter ses lments, puis de configurer et publier le modle.

Voir Modles de donnes [p 34].

3.3 Gestion des donnes

Section de la documentation Jeux de donnes [p 120]

enregistrement

Ensemble de donnes identifi de manire unique par une cl primaire. Un enregistrement correspond une ligne dans une table. Chaque enregistrement respecte la structure de donnes dfinie dans le modle de donnes associ. C'est ce modle de donnes qui indique les types et les cardinalits des champs qui composent l'enregistrement.

Voir table (jeu de donnes) [p 26], cl primaire [p 25].

L'ancien nom (avant la version 5) pour "enregistrement" tait "occurrence".

table (jeu de donnes)

Ensemble d'enregistrements (lignes) de mme structure contenant des donnes. Chaque enregistrement est identifi de manire unique par sa cl primaire.

Les tables sont reprsentes par l'icne .

Voir enregistrement [p 26], cl primaire [p 25].

jeu de donnes

Instance d'un modle de donnes qui contient les donnes. La structure et le comportement d'un jeu de donnes sont bass sur les dfinitions fournies par le modle de donnes qu'il implmente. En fonction de son modle de donnes, un jeu de donnes peut contenir des donnes sous la forme de tables, groupes et champs.

Voir table (jeu de donnes) [p 26], champ [p 25], groupe [p 25], vues [p 27].

Les jeux de donnes sont represents par l'icne .

Concept apparent : Jeux de donnes [p 120].

L'ancien nom (avant la version 5) de "jeu de donnes" tait "adaptation instance".

hritage

Mcanisme par lequel une donne d'une entit peut tre valorise par dfaut partir d'une autre entit. Dans EBX, deux types d'hritage sont possibles:le premier entre deux jeux de donnes, le deuxime entre deux champs.

Lorsqu'il est activ, l'hritage entre jeux de donnes permet un jeu de donnes enfant d'obtenir comme valeur par dfaut les donnes du jeu de donnes parent. Il est possible de surcharger dans les jeux de donnes enfants les valeurs hrites du parent. Par dfaut, l'hritage est dsactiv. Il peut tre activ lors de la dfinition du modle de donnes.

L'hritage depuis le jeu de donnes parent est reprsent par l'icne .

L'hritage de champ fonctionne de manire similaire, mais s'applique entre champs d'un mme jeu de donnes. Le champ hrit prend pour valeur par dfaut la valeur du champ source.

Les champs hrits sont reprsents par l'icne .

Concept apparent : Inheritance and value resolution [p 294].

vues

Configuration d'affichage applicable sur une table. Une vue peut tre cre pour un utilisateur ou un rle et permet de spcifier notamment sous quel mode seront affichs les enregistrements:hirarchique ou tabulaire ; ainsi que de dfinir des critres de filtrage et de tri.

Le mode de vue hirarchique prsente les donnes d'une table sous forme d'arbre. Cette vue est utilise pour montrer les relations entre les donnes du modle. Lors de la cration d'une vue hirarchique, une dimension doit tre slectionne pour dterminer les relations exploiter. Dans une vue hirarchique, il est possible de naviguer travers des relations rcursives, ainsi qu'entre plusieurs tables en utilisant des cls trangres.

Voir aussi

Vues [p 127]

Hirarchies [p 129]

vue recommande

Une vue recommande peut tre dfinie par le propritaire d'un jeu de donnes pour chaque profil cible. Quand un utilisateur se connecte sans spcifier de vue, la vue recommande, si elle existe, s'applique. Sinon, la vue par dfaut s'affiche.

L'action 'Grer les vues recommandes' permet de dfinir les rgles d'attribution des vues recommandes par utilisateur et par rle.

Concept apparent : Vues recommandes [p 131].

vue favorite

Lors de la consultation d'une table, l'utilisateur peut choisir de dfinir la vue actuelle comme sa vue favorite via le sous-menu 'Grer les vues'.

Une fois dfinie comme favorite, la vue s'appliquera automatiquement pour cet utilisateur lors de chaque accs la table.

Concept apparent : Grer les vues [p 132].

3.4 Cycle de vie des donnes

Section de la documentation Espaces de donnes [p 98]

espace de donnes

Contient les jeux de donnes. L'espace de donnes est utilis pour isoler diffrentes versions de jeux de donnes ou pour les organiser. Des espaces de donnes enfants peuvent tre crs partir d'un espace de donnes. Un espace de donnes enfant est initialis dans le mme tat que son parent au moment de sa cration. Ultrieurement, l'enfant pourra tre fusionn avec son parent. A tout moment, une comparaison avec d'autres espaces de donnes est possible.

Voir hritage [p 27], rfrentiel [p 23], fusion [p 28].

Concept apparent : Dataspaces [p 98].

L'ancien nom (avant la version 5) pour "espace de donnes" tait "branch" ou "snapshot".

espace de donnes de rfrence

Anctre commun des espaces de donnes du rfrentiel. N'ayant pas de parent, cet espace de donnes ne peut pas tre fusionn.

Voir espace de donnes [p 28], fusion [p 28], rfrentiel [p 23].

fusion

Intgration, dans l'espace de donnes parent, des changements raliss dans un espace de donnes enfant depuis sa cration. L'espace de donnes enfant est ferm aprs une fusion ralise avec succs. Pour effectuer cette fusion, un passage en revue des diffrences entre les deux espaces de donnes est requis afin de rsoudre les ventuels conflits. En effet, des conflits peuvent survenir en cas de modification des mmes donnes tant sur l'enfant que le parent. Une dcision doit tre prise pour chacun de ces conflits, afin de dterminer quelle modification doit prendre le pas sur l'autre.

Concept apparent : Fusion [p 107].

image

Copie statique d'un espace de donnes qui capture son tat et tout son contenu un moment donn, afin d'tre utilise comme rfrence. Une image peut tre consulte, exporte, et compare d'autres espaces de donnes, mais jamais modifie directement.

Les images sont reprsentes par l'icne .

Concept apparent : Image [p 113]

L'ancien nom (avant la version 5) pour "image" tait "version" ou "home".

3.5 Historique

Section de la documentation History [p 273]

historisation

Mcanisme qui peut tre activ au niveau d'une table afin de suivre les modifications dans le rfrentiel. Deux vues d'historique sont disponibles quand l'historisation est active : la vue historique de table et la vue historique des transactions. Dans toutes les vues d'historique, les fonctionnalits classiques des tables telles que l'export, la comparaison et les filtres restent disponibles.

L'activation de l'historique ncessite la configuration d'un profil d'historisation. L'historisation des tables n'est pas active par dfaut.

Voir vue historique de table [p 29], vue historique des transactions [p 29], profil d'historisation [p 29].

profil d'historisation

Ensemble de prfrences spcifiant d'une part les espaces de donnes dont les modifications doivent tre enregistres dans l'historique de la table et, d'autre part, si les transactions doivent chouer quand l'historisation n'est pas disponible.

Voir profil d'historisation [p 29].

vue historique de table

Vue contenant la trace de toutes les modifications effectues sur une table donne, notamment les crations, mises jour et suppressions. Chaque entre prsente les informations transactionnelles telles que:la date et l'heure, l'utilisateur ayant effectu l'action, ainsi que l'tat des donnes l'issue de la transaction. Ces informations peuvent aussi tre consultes au niveau d'un enregistrement ou d'un jeu de donnes.

Rfrence technique apparente History [p 273].

vue historique des transactions

Vue prsentant les donnes techniques et d'authentification des transactions au niveau du rfrentiel ou d'un espace de donnes. Etant donn qu'une transaction peut effectuer de multiples oprations/actions et peut affecter plusieurs tables dans un ou plusieurs jeux de donnes, cette vue montre toutes les oprations qui ont t effectues dans le primtre en question pour chaque transaction.

Rfrence technique apparente History [p 273].

3.6 Modlisation de workflow

Section de la documentation Modles de workflow [p 148]

modle de workflow

Dfinition de la succession d'oprations effectuer sur les donnes.

Un modle de workflow de donnes dcrit la totalit du parcours que doivent suivre les donnes pour tre traites, que ce soit en termes d'tats ou d'actions associes effectuer par des utilisateurs et des tches automatiques.

Concept apparent : Modles de workflow [p 148].

L'ancien nom (avant la version 5) de "modle de workflow" tait "workflow definition".

Les modles de workflows sont reprsents par l'icne .

tche automatique

Tche de workflow de donnes effectue par une procdure automatique, sans intervention humaine.

Les tches automatiques les plus communes sont la cration d'espace de donnes, la fusion d'espace de donnes et la cration d'image.

Les tches automatiques sont reprsentes par l'icne .

Voir modle de workflow [p 29].

tche utilisateur

Tche de workflow compose d'un ou plusieurs bons de travail raliss en parallle par des utilisateurs (intervention humaine).

Les bons de travail sont proposs ou assigns aux utilisateurs, en fonction du modle de workflow de donnes. L'avancement du workflow de donnes positionn sur une tche utilisateur dpend de la satisfaction du critre de fin de tche dfini dans le modle de workflow de donnes.

Les tches utilisateur sont reprsentes par l'icne .

Voir modle de workflow [p 29].

condition de workflow

Etape de dcision dans le workflow de donnes.

Une condition de workflow de donnes dcrit le critre utilis pour dterminer quelle sera la prochaine tape excuter.

Les conditions de workflow sont reprsentes par l'icne .

appel des sous-workflows

Etape qui met le workflow de donnes courant en attente et qui lance un ou plusieurs autres workflows de donnes. Si une telle tape lance plusieurs sous-workflows, les sous-workflows sont excuts en parallle.

tche d'attente

Etape d'un workflow de donnes qui met en pause le workflow en cours en attendant un vnement donn. Lorsque l'vnement est reu, le workflow est rveill et va automatiquement l'tape suivante.

contexte des donnes

Ensemble de donnes qui peuvent tre partages entre les tapes pendant toute la dure de vie d'un workflow afin de garantir la communication entre les tapes.

3.7 **Workflows de donnes**

Section de la documentation Workflows de donnes [p 176]

publication de workflow

Version particulire d'un modle de workflow de donnes qui est mise disposition des utilisateurs ayant les permissions ncessaires pour excuter des workflows.

L'ancien nom (avant la version 5) de "publication de workflow" tait "workflow".

workflow de donnes

Instance particulire d'un modle de workflow qui excute les tapes dfinies dans le modle de workflow de donnes (les tches utilisateur, les tches automatiques et les conditions).

Voir modle de workflow [p 29].

Concept apparent : Workflows de donnes [p 176].

L'ancien nom (avant la version 5) de "workflow de donnes" tait "workflow instance".

corbeille

Liste des workflows de donnes publis, affichs en fonction des permissions de l'utilisateur. Les utilisateurs qui ont la permission de lancer des workflows peuvent le faire partir de leur corbeille. Tous les bons de travail ncessitant une action de l'utilisateur sont affichs sous la publication de workflow associe dans la corbeille. De plus, si l'utilisateur est administrateur de workflows de donnes, il a la possibilit de voir leur tat dans sa corbeille et ainsi d'intervenir, si ncessaire, sur les workflows qu'il supervise.

Voir workflow de donnes [p 31].

bon de travail

Action unitaire d'une tche utilisateur qui doit tre ralise par un utilisateur.

Les bons de travail allous sont reprsents par l'icne .

Voir tche utilisateur [p 30].

jeton

Repre de position d'un workflow qui indique quelle tape courante est actuellement excute par un workflow de donnes. Les jetons sont utiliss durant l'avancement d'un workflow de donnes et sont uniquement visibles par les administrateurs du rfrentiel.

3.8 **Services de donnes**

Section de la documentation Services de donnes [p 198]

service de donnes

EBX partage les donnes de rfrence conformment l' Architecture Oriente Service en utilisant la technologie XML web service. Tous les services de donnes sont gnrs directement partir des modles ou services built-in. Ils peuvent tre utiliss pour accder une partie des fonctionnalits disponibles via l'interface utilisateur.

Les services de donnes d'EBX proposent :

- Un gnrateur WSDL pour les modles de donnes ainsi que pour les services built-in. Le fichier WSDL peut-tre produit indiffremment au travers de l'interface utilisateur ou du connecteur HTTP(S) pour un logiciel d'intgration ou une application cliente. Les oprations de services sont invoques par des messages XML communiqus au point d'entre d'EBX.
- Un connecteur SOAP, ou point d'entre pour les messages SOAP, permet aux systmes externes d'interagir avec le contenu du rfrentiel. Ce connecteur rpond aux demandes issues des WSDL produits par EBX. Aprs authentification, il accepte les messages XML et les excute selon les permissions de l'utilisateur authentifi.
- Un connecteur RESTful, ou point d'entre pour les oprations de slection, permet aux systmes externes d'interroger le contenu du rfrentiel EBX. Aprs authentification, il accepte la requte dfinie dans l'URL et l'excute selon les permissions de l'utilisateur authentifi.

lignage

Mcanisme de mise en place de profils de droits d'accs pour des services de donnes. Les profils de droits d'accs ainsi dfinis sont utiliss pour accder aux donnes via des interfaces WSDL.

Concept apparent : Gnrer un WSDL pour un lignage [p 203].

3.9 **Transverse**

noeud

Un noeud est un lment d'une arborescence ou d'un graphe. Dans EBX, 'Noeud' peut faire rfrence diffrents concepts selon le contexte d'utilisation :

- Dans le contexte du modle de workflow [p 29], un noeud est une tape du workflow ou une condition.
- Dans le contexte du modle de donnes [p 24], un noeud est un groupe, une table ou un champ.
- Dans le contexte des hirarchies [p 27], un noeud reprsente une valeur d'une dimension.
- Dans un arbre d'hritage de jeux de donnes [p 27], un noeud est un jeu de donnes.
- Dans un jeu de donnes [p 26], un noeud est le noeud du modle de donnes valu dans le contexte du jeu de donnes ou de l'enregistrement.

Modèles de données

CHAPITRE 4

Introduction aux modles de donnes

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Utilisation de l'interface utilisateur de la section Modles de donnes

4.1 Prsentation

La fonction d'un modle de donnes

La premire tape de toute gouvernance de donnes dans TIBCO EBX est le dveloppement d'un modle de donnes. Le but d'un modle de donnes est de dfinir la structure des donnes gres dans le rfrentiel en termes d'organisation, de types de donnes, et de relations smantiques. Une fois que le modle de donnes a t dfini et publi, il devient possible de crer des jeux de donnes partir de celui-ci.

Afin de dfinir un modle de donnes dans le rfrentiel, crez d'abord un nouveau modle de donnes, puis dfinissez sa structure et les proprits de ses lments (tables, champs et groupes). Le modle de donnes ainsi dfini doit tre publi pour devenir disponible. Les utilisateurs pourront crer des jeux de donnes partir de cette publication qui contiendront les donnes gres par le rfrentiel EBX.

Concepts de base utiliss dans la modlisation des donnes

Une comprhension des termes suivants est ncessaire pour commencer la cration de modles de donnes :

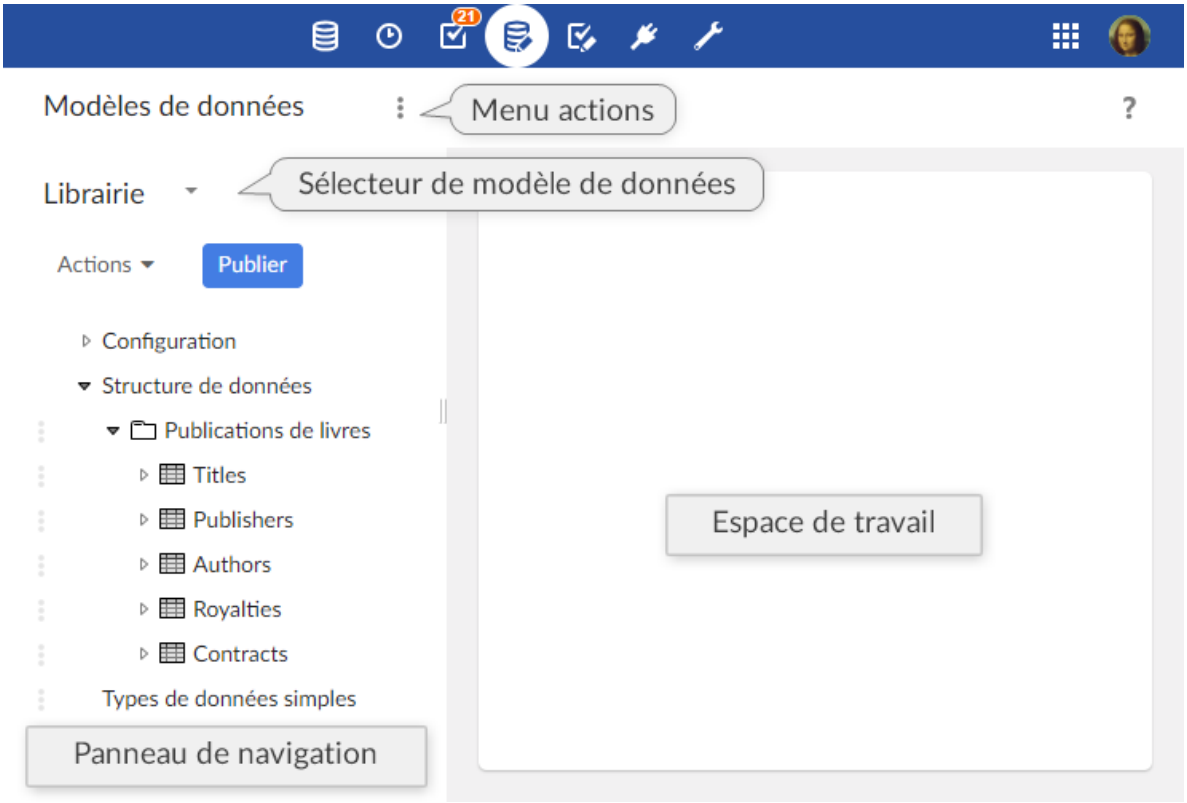
- champ [p 25]
- cl primaire [p 25]
- cl trangre [p 25]
- table [p 25]
- groupe [p 25]
- type rutilisable [p 26]
- rgle de validation [p 26]

4.2 Utilisation de l'interface utilisateur de la section Modles de donnes

Navigation dans le Data Model Assistant

Les modles de donnes peuvent tre imports, dits, et publis dans la section **Modles de donnes**. L'assistant fourni dans EBX permet d'laborer facilement des modles de donnes.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance'.



Le panneau de navigation est organis selon les sections suivantes :

Configuration	La configuration technique du modle de donnes.
Proprits globales	Dfinit les proprits techniques du modle.
Modles de donnes inclus	Dfinit les modles de donnes inclus dans le modles courant. Tous les types dfinis par les modles de donnes inclus pourront tre utiliss dans le modle courant.
'Bindings' du modle	Les bindings du modle d'adaptation spcifient quels sont les types Java gnrer partir du modle.
Librairie de composants	Dfinit les composants Java disponibles dans le modle. Ces composants reprsentent l'ensemble des fonctionnalits programmatiques qui seront disponibles dans le modle (contraintes programmatiques, fonctions, bean, etc.).
Barres d'outils	Les barres d'outils disponibles dans le modle de donnes.
Composants Ajax	Dfinit les composants Ajax disponibles dans le modle.
Services utilisateurs	Dclare les services utilisateurs utilisant l'API disponible avant la version 5.8.0. A partir de la version 5.8.0, il est recommand d'utiliser la nouvelle API UserService (ces services tant directement enregistrs via l'API Java, il n'est plus ncessaire de les dclarer dans le Data Model Assistant).
Add-ons	Dfinit les add-ons utiliss par le modle de donnes. Ces add-ons pourront enrichir le modle de donnes aprs la publication en ajoutant des proprits et contraintes aux lments du modle.
Services de donnes	Cette table dfinit les suffixes des oprations WSDL qui peuvent tre utiliss dans le modle de donnes. Ces suffixes permettent, dans le cadre d'une opration d'un service de donnes, de faire rfrence une table en utilisant un nom unique au lieu de son chemin dans le modle de donnes.
Rplications	Cette table dfinit les units de rplication du modle de donnes. Une unit de rplication permet la rplication d'une table source dans la base de donnes relationnelle, de sorte que les systmes externes peuvent accder ces donnes par le biais de requtes et de vues SQL.

Structure de donnes	Structure du modle de donnes. Dfinit les relations entre les lments du modle de donnes et permet d'accder la dfinition de chaque lment.
Types de donnes simples	Types simples rutilisables dfinis dans le modle de donnes courant.
Types de donnes complexes	Types complexes rutilisables dfinis dans le modle de donnes courant.
Types de donnes simples inclus	Types simples rutilisables dfinis dans un modle de donnes inclus dans le modle courant.
Types de donnes complexes inclus	Types complexes rutilisables dfinis dans un modle de donnes inclus dans le modle courant.

Voir aussi

Modlisation de la structure des donnes *[p 49]*

Configuration du modle de donnes *[p 41]*

Types rutilisables *[p 51]*

Icnes des lments du modle de donnes

champ [p 25]

cl primaire [p 25]

cl trangre [p 25]

table [p 25]

groupe [p 25]

Concepts apparentés

Espaces de donnes *[p 98]*

Jeux de donnes *[p 120]*

CHAPITRE 5

Cration du modle de donnes

Ce chapitre contient les sections suivantes :

1. Cration d'un modle de donnes
2. Slection d'un type de modle de donnes

5.1 Cration d'un modle de donnes

Pour crer un modle de donnes, cliquez sur le bouton **Crer** dans le slecteur, puis suivez les instructions de l'assistant de cration de modle de donnes.

5.2 Slection d'un type de modle de donnes

Si l'utilisateur a le rle "Administrateur", il pourra slectionner le type de modle de donnes. Un utilisateur avec le rle "Administrateur" peut crer un modle de donnes *smantique* ou *relationnel*.

Modles smantiques

Les modles smantiques permettent l'utilisation de toutes les fonctionnalits de gestion des donnes fournies par TIBCO EBX, comme la gestion du cycle de vie, en utilisant des espaces de donnes. Par dfaut, les modles de donnes dans EBX sont smantiques.

Modles relationnels

Les modles relationnels sont utiliss lorsque les tables du modle doivent tre accessibles par le biais d'un systme de gestion de base de donnes (SGBD). Le principal avantage des modles relationnels est la possibilit d'interroger les tables par des requtes SQL externes. Cependant, les modles relationnels perdent certaines fonctionnalits de gestion des donnes, par exemple l'hritage, les champs multi-valus, et la gestion de cycle de vie par l'utilisation des espaces de donnes.

> **Note**
>
> Un modle de donnes relationnel peut tre utilis par un seul jeu de donnes. L'espace de donnes contenant le jeu de donnes doit galement tre dclar comme relationnel.

Voir aussi

Relational mode *[p 267]*

Espaces de donnes *[p 98]*

CHAPITRE 6

Configuration du modle de donnes

Ce chapitre contient les sections suivantes :

1. Informations associes au modle de donnes
2. Permissions
3. Proprits du modle de donnes
4. Modles de donnes inclus
5. Services de donnes
6. Rplication des donnes vers tables relationnelles
7. Add-ons utiliss par le modle de donnes

6.1 Informations associes au modle de donnes

Pour visualiser et diter les informations concernant le propritaire et la documentation du modle de donnes, slectionnez "Informations" dans le menu **"Actions" du modle de donnes** [p 35] dans le panneau de navigation.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance'.

Nom unique	Le nom unique du modle de donnes. Ce nom ne peut pas tre modifi aprs la cration du modle.
Propritaire	Spcifie le propritaire du modle de donnes, qui a le droit de modifier les informations du modle et ses permissions.
Documentation localise	Libells et descriptions localiss pour le modle de donnes.

6.2 Permissions

Pour dfinir les permissions d'accs au modle de donnes, slectionnez "Permissions" dans le menu **"Actions" du modle de donnes** [p 35] dans le panneau de navigation.

La dfinition des permissions d'un modle de donnes s'effectue de la mme manire que pour les jeux de donnes. Les dtails se trouvent dans la section Permissions [p 140].

6.3 Proprits du modle de donnes

Dans le panneau de navigation, sous Configuration > Proprits du modle, vous pouvez accder aux proprits techniques suivantes :

Nom du module	Dfinit le module contenant les ressources qui seront utilises par ce modle de donnes. Ce nom de module dsigne galement le module cible lors de la publication du modle de donnes s'il est publi dans un module.
Chemin du module	Localisation physique du module sur le systme de fichiers du serveur.
Chemins des sources	Emplacements des codes sources utiliss pour configurer les composants Java dans la "Librairie de composants". Si un chemin est relatif, il sera rsolu partir du "Chemin du module".
Mode de publication	Les deux modes possibles sont:la publication du modle dans un document XML Schema au sein d'un module ou en tant que modle de donnes embarqu dans le rfrentiel TIBCO EBX. Les modles de donnes embarqus permettent d'avoir des fonctionnalits additionnelles, comme la gestion des versions et la restauration des versions prcdentes du modle. Voir Modes de publication [p 91] pour plus d'informations. **Chemin du modle dans le module** : Dfinit l'emplacement du document XML Schema pour la publication du modle de donnes dans un module. Le chemin doit commencer par "/".
Hritage des donnes	Spcifie si l'hritage des donnes, entre jeux de donnes, est activ pour ce modle de donnes. L'hritage des donnes est dsactiv par dfaut. Voir hritage [p 143] pour plus d'informations.
Documentation	Documentation du modle de donnes dfinie l'aide d'une classe Java. Cette classe Java dfinit les libells et descriptions des lments du modle de donnes. Les libells et descriptions dfinis par cette classe Java sont affichs, dans les jeux de donnes associs, en priorit par rapport aux libells et descriptions dfinis localement par les lments du modle de donnes. Voir Dynamic labels and descriptions [p 556] pour plus d'informations.

Extensions spciales	Permissions dfinies l'aide de rgles programmatiques.
Dsactiver les contrles de l'auto-incrment	Indique si le contrle de la valeur d'un champ auto-incrment par rapport la valeur maximale trouve dans la table en cours de mise jour doit tre dsactiv. Voir Valeurs auto-incrmentes [p 553] pour plus d'informations.
Activer services utilisateurs (ancienne API)	Indique si des services utilisateurs utilisant l'API disponible avant la version 5.8.0 peuvent tre dclars. Si 'Non', la section "Configuration > Services utilisateurs" n'est pas affiche (sauf si cette section dclare dj au moins un service). A partir de la version 5.8.0, il est recommand d'utiliser l'API UserService (ces services tant directement enregistrs via l'API Java, il n'est plus ncessaire de les dclarer dans le Data Model Assistant). Voir `UserServiceDeclaration`[API] pour plus d'informations.

6.4 Modles de donnes inclus

Les types de donnes dfinis dans un autre modle de donnes peuvent tre utiliss dans le modle de donnes courant en ajoutant une entre pour l'autre modle de donnes dans la table Configuration > Modles de donnes inclus.

En accdant l'enregistrement du modle inclus depuis cette table, des informations techniques lies ce modle sont consultables sous l'onglet **Information**. Cet onglet contient aussi le rapport de validation du modle de donnes inclus.

Seuls les modles de donnes sans erreur de validation, qui ont t dfinis et publis comme modle "embarqu" ou dans un module, peuvent tre inclus.

Les types de donnes doivent tre uniques la fois dans le modle courant et dans tous les modles inclus. Il est impossible d'inclure un modle contenant des types de donnes dj existant dans le modle courant ou dans les autres modles de donnes inclus.

Voir aussi*Including external data models [p 515]*

6.5Services de donnes

Il est possible de faire rfrence, dans une opration d'un service de donnes, une table en utilisant un nom d'entit unique au lieu de son chemin dans le modle de donnes en dfinissant des suffixes pour les oprations WSDL. Un suffixe WSDL est l'association entre le chemin d'une table et un nom.

Pour dfinir un suffixe WSDL en utilisant l'interface utilisateur, crer un nouvel enregistrement dans la table 'Services de donnes' situe dans la section 'Configuration du modle de donnes' dans le panneau de navigation. Un enregistrement de cette table dfinit les proprits suivantes :

Chemin de la table	Indique le chemin de la table dans le modle de donnes qui doit tre associe ce nom d'entit.
Suffixe d'opration WSDL	Ce nom est utilis pour suffixer tous les noms des oprations qui concernent la table spcifie. Si cette proprit n'est pas dfinie pour une table donne, alors le dernier lment du chemin de la table est utilis. Ce nom doit tre unique dans le modle de donnes.

Voir aussi*Data services* *[p 569]*

6.6 Rplication des donnes vers tables relationnelles

Dans n'importe quel type de modle de donnes, il est possible de dfinir des *units de rplication* afin que les donnes du rfrentiel soient copies dans des tables relationnelles ddies. Ainsi, ces tables relationnelles permettent d'accder directement aux donnes en utilisant des requtes et des vues SQL.

Pour dfinir une unit de rplication en utilisant l'interface utilisateur, crez un nouvel enregistrement dans la table "Rplications" situe dans la section Configuration du modle de donnes dans le panneau de navigation. Chaque unit de rplication concerne un jeu de donnes spcifique dans un espace de donnes

particulier. Une unit de rplication peut inclure plusieurs tables, tant qu'elles sont dans le mme jeu de donnes. Une unit de rplication dfinit les informations suivantes :

Nom	Nom de l'unit de rplication. Ce nom identifie l'unit de rplication dans le modle de donnes. Ce nom doit tre unique.
Espace de donnes	Indique l'espace de donnes concern par la rplication. Cet espace de donnes ne peut ni tre une version ni tre relationnel.
Jeu de donnes	Indique le jeu de donnes concern par la rplication.
Mode de rafraichissement	Dfinit le mode de synchronisation. Les diffrents modes de synchronisation sont les suivants: • **Sur "commit"**:les donnes rpliques contenues dans la base de donnes sont toujours jour par rapport la table source. Chaque transaction de mise jour de la table source produit les mises jour correspondantes dans la table contenant les donnes rpliques dans la base de donnes. • **Sur demande**:les donnes rpliques contenues dans la base de donnes sont mises jour uniquement lorsqu'une opration manuelle de rafraichissement est effectue.
Tables	Indique les tables du modle de donnes qui doivent tre rpliques dans la base de donnes. **Chemin de la table**:Indique le chemin de la table dans le modle de donnes qui doit tre rplique dans la base de donnes. **Nom dans la base de donnes**:Indique le nom de la table dans la base de donnes qui contiendra les donnes rpliques. Ce nom doit tre unique par rapport toutes les units de rplication.
Listes agrges	Dfinit les proprits des listes agrges contenues dans la table. **Chemin de l'lment**:Indique le chemin de la liste agrge dans la table qui doit tre rplique dans la base de donnes. **Nom dans la base de donnes**:Indique le nom de la table dans la base de donnes qui contiendra les donnes rpliques de la liste agrge. Ce nom doit tre unique par rapport toutes les units de rplication.

Voir aussi*Rplication [p 281]*

6.7 **Add-ons utiliss par le modle de donnes**

Dans tout type de modle de donnes, il est possible de dfinir les *add-ons* utiliss dans le modle de donnes courant. Ces add-ons pourront enrichir le modle de donnes aprs la publication en ajoutant des proprits et contraintes aux lments du modle.

Pour dfinir un add-on en utilisant l'interface utilisateur, crer un nouvel enregistrement dans la table 'Add-ons' situe dans la section 'Configuration du modle de donnes' dans le panneau de navigation. Un enregistrement de cette table dfinit les proprits suivantes :

Nom de l'add-on	Description de l'add-on.
Version de l'add-on.	Indique la version de l'add-on.
Activ	Indique si l'add-on est activ. Un add-on doit tre activ pour tre utilis.

CHAPITRE 7

Modlisation de la structure des donnes

Pour dfinir la structure du modle de donnes, slectionnez le modle de donnes avec lequel vous voulez travailler dans le panneau de navigation.

La structure du modle de donnes est accessible depuis le panneau de navigation dans la section "Structure de donnes". Cette section permet de visualiser et de dfinir la structure des champs, groupes, et tables du modle de donnes.

Ce chapitre contient les sections suivantes :

1. Actions et proprits communes
2. Types rutilisables
3. Dtails de la cration des lments du modle de donnes
4. Modification des lments existants

7.1 Actions et proprits communes

Crer des lments

Les lments suivants peuvent tre ajouts un modle de donnes :

- champs
- groupes
- tables
- cls primaires
- cls trangres

- associations

Ajoutez un de ces lments sous un lment existant en cliquant sur la flche situe la droite de l'lment existant, puis en slectionnant une option de cration parmi les options prsentes dans le menu. Suivez ensuite l'assistant de cration pour crer un lment.

> **Note**
>
> L'lment `root` est ajout par dfaut lors de la cration d'un modle de donnes. Cet lment reprsente la racine de la structure du modle de donnes. S'il faut renommer cet lment, il peut tre supprim et recr avec un nom diffrent.

Noms, libells, descriptions, et informations

Le nom de l'lment crer est obligatoire. Ce nom doit tre unique au sein d'un mme niveau dans la structure de donnes. En effet, sous un mme groupe, deux lments ne peuvent avoir le mme nom. Une fois l'lment cr, son nom ne peut plus tre modifi.

Il est possible de dfinir des libells localiss qui seront affichs dans l'interface utilisateur au lieu du nom unique de l'lment. Il est aussi possible de dfinir une description localise de l'lment. Contrairement au nom de l'lment, les libells et descriptions sont modifiables aprs la cration de l'lment. Selon la prfrence de langue de chaque utilisateur, TIBCO EBX affichera le libell et la description localiss de l'lment.

Supprimer des lments

Tous les lments du modle de donnes peuvent tre supprims de la structure de donnes en utilisant la flche situe la droite de l'lment supprimer.

Si un groupe ou une table n'utilisant pas un type rutilisable est supprim, la suppression est effectue rcursivement sur tous les lments situs sous le groupe ou la table.

Dupliquer des lments existants

Pour dupliquer un lment, cliquez sur la flche situe la droite de l'lment dupliquer. Spcifiez le nom de l'lment dupliqu. Ce nom doit tre unique au sein d'un mme niveau dans la structure de donnes. Toutes les proprits de l'lment source sont dupliques.

L'lment dupliqu est rajout dans le modle de donnes au mme niveau que l'lment d'origine, en dernire position. Lorsqu'un lment contenant d'autres lments est dupliqu, tous les sous-lments sont dupliqus avec leurs noms originaux.

> **Note**
>
> En cas de duplication d'un champ appartenant une cl primaire, les proprits du champ sont dupliques mais le nouveau champ n'est pas ajout la cl primaire de la table parente.

Dplacer des lments

Pour dplacer un lment, cliquez sur la flche situe la droite de l'lment dplacer et slectionnez "Dplacer". Slectionnez ensuite la flche qui correspond l'lment *avant lequel* positionner l'lment actuel.

> **Note**
>
> Le dplacement d'un lment est uniquement possible au sein d'un mme niveau dans la structure de donnes du modle.

7.2 **Types rutilisables**

Les types rutilisables sont des lments partags qui, aprs leur cration, peuvent tre rutiliss par diffrents lments du modle de donnes.

> **Note**
>
> En modifiant la dfinition d'un type rutilisable, la structure de tous les lments bass sur ce type rutilisable est aussi modifie. L'arborescence "Structure de donnes" affiche, en lecture seule, la structure d'un groupe ou d'une table qui utilise un type rutilisable. Pour diter la structure du type rutilisable associ, accdez au type dans la section "Types de donnes simples" ou "Types de donnes complexes".

Dfinition d'un type rutilisable

En utilisant le menu avec la flche des sections "Types de donnes simples" ou "Types de donnes complexes" dans le panneau de navigation, il est possible de crer des types simples et des types complexes rutilisables qui seront disponibles pour crer d'autres lments avec la mme structure et les mmes proprits. Il est galement possible de convertir les tables et groupes existants en types rutilisables

en utilisant le menu avec la flche situ ct de l'lment convertir.

Il est possible de visualiser les lments du modle utilisant un type rutilisable, en ditant ce type et en cliquant sur le lien "Rfrences vers ce type". Ce lien affiche une table listant tous les lments utilisant ce type. Si le type n'est utilis par aucun lment, il peut tre supprim en slectionnant "Supprimer type"

en utilisant le menu avec la flche situ droite du type supprimer.

Utilisation d'un type rutilisable

Les structures et les proprits de nouveaux lments peuvent tre dfinies par des types rutilisables en slectionnant un type rutilisable la cration d'un lment. L'lment cr utilisera la structure et les proprits du type rutilisable.

Inclusion des types de donnes dfinis dans d'autres modles de donnes

Les types rutilisables peuvent aussi tre partags entre plusieurs modles de donnes. En configurant l'inclusion d'un modle de donnes externe, il est possible d'utiliser les types de donnes inclus pour crer des lments dans la structure de donnes, de la mme manire que pour les types rutilisables dfinis en local.

> **Note**
>
> Les types de donnes devant tre uniques pour tous les types dfinis en local et inclus, il n'est pas possible de crer un type rutilisable portant le mme nom qu'un type de donnes dans un modle de donnes inclus. De la mme manire, il n'est pas possible d'inclure un modle de donnes externe qui dfinit un type de donnes portant le mme nom qu'un type rutilisable dfini en local ou dans un autre modle de donnes inclus.

Les types de donnes inclus apparaissent dans les sections "Types de donnes simples inclus" et "Types de donnes complexes inclus" dans le panneau de navigation. Les dtails de ces types rutilisables sont consultables, mais ils ne sont ditables que dans leurs modles de donnes d'origine.

Voir Modles de donnes inclus [p 44] pour plus d'informations.

7.3 Dtails de la cration des lments du modle de donnes

Cration de champs

A la cration d'un champ, un type de donnes doit tre slectionn. Il dfinira le type de donnes associ aux valeurs saisies dans un jeu de donnes bas sur ce modle. Le type de donnes du champ ne peut pas tre modifi aprs la cration du champ.

Durant la cration d'un champ, il est galement possible de le dsigner comme cl trangre, champ obligatoire, ou comme cl primaire si le champ est cr sous une table.

Cration de tables

Lors de la cration d'une nouvelle table, un type rutilisable existant peut tre utilis pour dfinir la structure et les proprits de cette nouvelle table. Voir Types rutilisables [p 51] pour plus d'informations.

Chaque table ncessite la dsignation d'au moins un champ cl primaire, qui peut tre cr comme un lment enfant de la table dans la section "Structure de donnes" du panneau de navigation.

Cration de groupes

Lors de la cration d'un groupe, il est possible d'utiliser un type rutilisable existant pour dfinir la structure et les proprits du nouveau groupe. Voir Types rutilisables [p 51] pour plus d'informations.

Cration de cls primaires

Pour chaque table, il est ncessaire de dfinir une cl primaire. Pour cela, ajoutez un nouvel lment enfant partir du menu d'actions disponible sur la table dans la section "Structure de donnes" du panneau de navigation.

Il est aussi possible d'ajouter un champ existant la dfinition de la cl primaire, sur l'onglet "Cl primaire" dans les "Proprits avances" de la table.

Cration ou dfinition de cls trangres

Les champs associs une cl trangre sont de type "Chaîne de caractres". Pour crer une cl trangre sur une table, ajoutez un nouvel lment enfant partir du menu d'actions disponible sur la table dans la section "Structure de donnes" du panneau de navigation. Il est galement possible de dfinir directement les proprits d'une cl trangre en ditant un champ de type "Chaîne de caractres". Pour convertir un champ existant de type "Chaîne de caractres" en cl trangre, activez la proprit "Contrainte de cl trangre" dans les "Contrles avancs" du champ et dfinissez les proprits associes.

Il faut toujours spcifier la table rfrence par une cl trangre.

Cration d'associations

Une association permet de dfinir un lien smantique entre deux tables. Une association peut tre dfinie en crant un lment dans une table de la section "Structure de donnes" du panneau de navigation en slectionnant la proprit "association" dans le formulaire de cration. Une association peut uniquement tre cre dans une table et il n'est pas possible de convertir un champ existant en association.

Lors de la cration d'une association, spcifiez son type. Pour cela, diffrentes options sont disponibles:

- Relation inverse d'une *cl trangre*. Dans ce cas, l'association est dfinie dans une *table source* et fait rfrence une *table cible*. Ce type d'association est l'inverse d'une cl trangre qui est dfinie dans la table cible et qui rfrence la table source. Dfinissez la cl trangre qui rfrence la table contenant l'association. Pour cela, des assistants de saisie sont disponibles lors de la cration de l'association.
- Utilisation d'une *table de liens*. Dans ce cas, l'association est dfinie dans une *table source* et fait rfrence une *table cible* qui est infre partir d'une *table de liens*. Cette table de liens doit dfinir deux cls trangres:une cl trangre rfrenant la table source et une autre rfrenant la table cible. La cl primaire de la table de liens doit aussi tre compose de champs auto-incrments et/ou de cls trangres vers la table source ou cible de l'association. Dfinissez la table de liens et ces deux cls trangres. Pour cela, des assistants de saisie sont disponibles la cration de l'association.
- Utilisation d'un *prdicat XPath*. Dans ce cas, l'association est dfinie dans une *table source* et fait rfrence une *table cible* spcifi par un chemin *XPath*. Une *expression XPath* doit aussi tre dfinie afin de spcifier les critres qui permettent d'associer un enregistrement de la table courante avec des enregistrements de la table cible.

Pour ces types d'association, nous appelons *enregistrements associs* les enregistrements de la table cible smantiquement lis aux enregistrements de la table source.

Une fois l'association cre, il est possible de dfinir d'autres proprits:

- filtrer les enregistrements associs en dfinissant un filtre XPath. Il est uniquement possible d'utiliser les champs de la table source et de la table cible lors de la dfinition du filtre XPath. Il n'est donc pas possible d'utiliser les champs d'une table de liens dans un filtre XPath. Un assistant de saisie est disponible pour dfinir les champs utiliser dans un filtre XPath.
- configurer une vue tabulaire pour dfinir les champs de la table cible devant tre affichs dans les formulaires des jeux de donnes. Il n'est pas possible de configurer ou de modifier une vue tabulaire si la table cible n'est pas dfinie ou n'existe pas. Par dfaut, tous les champs de la table cible qu'un utilisateur a le droit de voir seront affichs dans les jeux de donnes si la vue tabulaire n'est pas dfinie.
- dfinir comment doivent tre prsents les enregistrements associs dans les formulaires des jeux de donnes. Indiquez si les enregistrements associs doivent tre inclus dans le formulaire ou dans un

onglet spcifique. Par dfaut, les enregistrements associs seront inclus dans le formulaire au niveau de l'association dans le modle de donnes.

- inclure / exclure les enregistrements associs dans les oprations de slection de Data Service. Par dfaut, les enregistrements ne sont pas inclus dans les oprations de slection des Data Service.
- spcifier les nombres minimum et maximum d'enregistrements associs requis. Dans les jeux de donnes associs, un message de validation est ajout lorsque les nombres minimum ou maximum d'enregistrements associs ne correspondent pas ces critres. Par dfaut, les nombres minimum et maximum d'enregistrements associs requis ne sont pas restreints.
- dfinir une contrainte de validation en utilisant un prdicat XPath pour restreindre les enregistrements associs. Il est uniquement possible d'utiliser les champs de la table source et de la table cible lors de la dfinition du prdicat XPath. Il n'est donc pas possible d'utiliser les champs d'une table de liens dans un prdicat XPath. Un assistant de saisie permet de slectionner les champs utiliser dans un prdicat XPath. Dans les jeux de donnes associs, un message de validation sera ajout pour tout enregistrement associ ne vrifiant pas cette contrainte.

7.4 Modification des lments existants

Suppression d'un champ de la cl primaire

Tout champ appartenant la cl primaire peut tre supprim de la cl primaire d'une table sur l'onglet "Cl primaire" dans les "Proprits avances" de la table.

Voir cl primaire [p 25] dans le glossaire.

CHAPITRE 8

Proprits des lments du modle de donnes

Aprs avoir cr un lment, il est possible de dfinir des proprits supplmentaires pour complter sa dfinition.

Voir aussi*Contrles sur les lments du modle de donnes* *[p 71]*

Ce chapitre contient les sections suivantes :

1. Proprits basiques des lments
2. Proprits avances des lments

8.1 Proprits basiques des lments

Proprits basiques communes

Les proprits basiques suivantes sont disponibles pour plusieurs types d'lments :

Information	Informations supplmentaires non internationalises associes l'lment.
Nombre minimum de valeurs	Nombre minimum de valeurs pour cet lment. Les cls primaires ne pouvant tre multi-values, cette proprit doit tre gale "1" ou tre "non dfinie". Pour les lments de type "Noeud de slection", le nombre minimum est automatiquement dfini "0".
Nombre maximum de valeurs	Nombre maximum de valeurs pour cet lment. Si cette proprit est suprieure "1", l'lment est considr comme multi-valu. Les cls primaires ne pouvant tre multi-values, cette proprit doit tre gale "1" ou tre "non dfinie". Pour une table, le nombre maximum d'lments est dfini "unbounded" par dfaut lors de sa cration. Pour les lments de type "Noeud de slection",le nombre maximum d'lments est dfini "0" par dfaut lors de la dfinition des proprits du noeud de slection.
Rgles de validation	Cette proprit est disponible pour les champs situs dans une table, sauf pour les champs de type `Mot de passe`, les types rutilisables, les champs des types complexes rutilisables et les noeuds de slection. Cette proprit est utilise pour dfinir des rgles de validation riches et complexes l'aide d'un diteur de prdicats XPath 1.0. Voir diteur de prdicats [p 317]. Une rgle de validation peut tre utile lorsque la validation de la valeur dpend de critres complexes ou des valeurs d'autres champs. Il est aussi possible d'indiquer qu'une rgle de validation dfinit des conditions sous lesquelles la valeur d'un champ est obligatoire. La valeur du champ est obligatoire si les critres de la rgle ne sont pas respects. Voir Contraintes sur valeurs "null" [p 545] pour plus d'informations. En utilisant l'assistant associ, il est possible de dfinir des libells localiss pour la rgle de validation, ainsi qu'un

message localis avec la svrit qui s'affichera si le critre n'est pas satisfait.

Il est possible d'indiquer si la rgle doit toujours tre valide ou non lorsqu'un utilisateur soumet un formulaire. Cette option est uniquement disponible sur les rgles de validation dfinies par un champ et lorsque la svrit du message est dfinie "erreur". Si les erreurs de validation ne sont pas autorises, alors toutes les saisies qui rendraient la rgle invalide seront rejetes et les donnes seront non modifies. Si les erreurs de validation sont autorises alors toutes les saisies qui rendraient la rgle invalide seront acceptes et les donnes seront modifies. Si cette proprit n'est pas spcifie, alors la rgle bloquera uniquement les erreurs lors de la soumission d'un formulaire.

Si une rgle de validation est dfinie sur un champ de type table, alors cette rgle sera considre comme une "contrainte sur table" et sera excute sur chaque enregistrement de la table. Voir Contraintes sur table [p 545] pour plus d'informations.

Proprits basiques des champs

Les proprits basiques suivantes sont spcifiques aux champs simples :

Valeur par dfaut	Dfinit une valeur par dfaut pour ce champ. Cette valeur sera insre automatiquement dans le champ de saisie du formulaire de cration des nouveaux enregistrements. Le type de la valeur par dfaut doit tre compatible avec le type du champ courant. Voir Valeur par dfaut [p 561] pour plus d'informations.
Message d'erreur de conversion	Messages d'erreur internationaliss affichs aux utilisateurs lors de la saisie d'une valeur qui n'est pas compatible avec le type du champ courant.
Rgle de calcul	Cette proprit est disponible pour les champs des tables, except pour les types rutilisables. Dfinit une rgle de calcul pour la valeur du champ avec l'aide d'un diteur de prdicats XPath 1.0. Voir Editeur de prdicats [p 317] pour plus d'informations. Une rgle de calcul peut tre utile lorsqu'une valeur dpend d'autres valeurs dans le mme enregistrement, mais qu'un calcul programmatique n'est pas ncessaire. Les limitations suivantes existent pour les rgles de calcul : • Les rgles de calcul s'utilisent uniquement avec les champs simples d'une table. • Les rgles de calcul ne peuvent pas tre dfinies sur des champs du type `OResource` ou `Password`. • Les rgles de calcul ne peuvent pas tre dfinies sur les noeuds de slection ni les champs cls primaires. • Les rgles de calcul ne peuvent pas tre dfinies en accdant l'lment depuis le rapport de validation.

8.2 **Proprits avances des lments**

Proprits avances communes

Les proprits avances suivantes sont disponibles pour plusieurs types d'lments :

Affichage par dfaut et outils > Visibilit	Cette section permet d'indiquer si cet lment doit tre affich dans la vue par dfaut d'un jeu de donnes, dans l'outil de recherche textuelle d'un jeu de donnes, ou dans l'opration "select" d'un service de donnes. Ces proprits sont ignores si elles sont dfinies sur un lment qui n'est pas dans une table. • Vue dirige par le modle Indique si l'lment courant doit tre affich ou non dans la vue par dfaut d'un jeu de donnes, la vue tabulaire par dfaut d'une table et le formulaire par dfaut associ aux enregistrements d'une table. La vue par dfaut d'un jeu de donnes, la vue tabulaire par dfaut d'une table, et le formulaire par dfaut associ aux enregistrements d'une table respectent la structure du jeu de donnes ou de la table dfinie dans le modle de donnes. Lorsque la proprit "Cach" est slectionnne, l'lment courant ne sera pas affich dans la vue par dfaut d'un jeu de donnes si l'lment courant est une table. Lorsque la proprit "Cach" est slectionnne, l'lment courant ne sera pas affich dans la vue tabulaire par dfaut et le formulaire par dfaut associ aux enregistrements d'une table si l'lment courant est dans une table. L'lment courant sera cependant affich dans l'assistant de cration de vues tabulaires et hirarchiques. Il sera donc possible de crer une vue personnalise qui affichera l'lment courant. Si cette proprit n'est pas renseigne, alors l'lment courant sera affich par dfaut. Cette proprit est ignore si elle est dfinie sur un champ qui n'est ni une table, ni dans une table. • Toutes les vues Lorsque la proprit 'Cach dans les vues' est slectionne, l'lment courant ne sera pas affich dans les vues par dfaut, tabulaires (vue par dfaut incluse) ou hirarchiques, d'une table d'un jeu de donnes. L'lment courant ne sera pas affich dans l'assistant de cration de vues tabulaires et hirarchiques. Il ne sera donc pas possible de crer une vue personnalise qui affichera l'lment courant. Cette proprit concerne uniquement les vues. L'lment courant sera donc affich dans le formulaire d'un enregistrement si la proprit 'Cach dans les vues' est slectionne. Cette proprit est ignore si elle est dfinie sur un champ hors d'une table.

- Outils de recherche

  Lorsque la proprit "Cach dans toutes les recherches" est slectionne, l'lment courant ne sera pas affich dans les outils de recherche textuelle et type d'un jeu de donnes. Lorsque la proprit "Cach uniquement dans une recherche textuelle" est slectionne, l'lment courant ne sera pas affich dans l'outil de recherche textuelle d'un jeu de donnes. Si ces proprits concernant les recherches ne sont pas renseignes, alors l'lment courant sera affich par dfaut dans les outils de recherche textuelle et type d'un jeu de donnes. Cette proprit est ignore si elle est dfinie sur un champ hors d'une table.

- Services de donnes

  Lorsque la proprit "Exclu dans les Data Services" est slectionnne, l'lment courant ne sera pas inclus lors de l'opration "select" d'un data service. Cette proprit est ignore si elle est dfinie sur un champ hors d'une table.

Voir Default view [p 563] dans le Guide du dveloppeur.

Affichage par dfaut et outils > Widget

Dfinit le widget utiliser. Un widget est un composant de saisie qui est affich dans les formulaires contenus dans les jeux de donnes associs. Si aucun widget n'est dfini, alors un widget par dfaut sera affich dans les jeux de donnes associs. Ce widget par dfaut est dfini partir du type et des proprits de l'lment courant. Il est possible d'utiliser un widget natif ou un widget personnalis. Un widget personnalis est dfini en utilisant une API Java. Cette API Java permet de dvelopper des composants de saisie riche pour des champs ou des groupes. Les widgets natifs et personnaliss ne peuvent pas tre dfinis sur une table et une association. Il est interdit de dfinir en mme temps un widget personnalis et un UI bean. Sur un champ de type cl trangre, il est interdit de dfinir en mme temps un widget personnalis et un slecteur de combo box.

Voir `UIWidgetFactory`[API] pour plus d'informations.

Affichage par dfaut et outils > Slecteur combo box

Dfinit le nom de la vue publie utilise dans le menu de slection de la cl trangre. Un bouton de slection sera affich en bas droite de la liste droulante du menu de slection. Lors de la dfinition d'une cl trangre, cette fonctionnalit permet d'accder une vue de slection avance en cliquant sur le bouton 'Slecteur' qui ouvre la vue de slection avance, permettant ainsi d'utiliser les options de tri et de recherche. Si aucune vue publie n'a t dfinie, la vue de slection avance sera dsactive. Si le chemin de la table rfrence est absolu, alors seules les vues publies correspondant cette table seront affiches. Si le chemin de la table rfrence est relatif,

alors toutes les vues associes au modle de donnes contenant la table cible seront affiches. Cette proprit ne peut pas tre utilise si un widget personnalis est dfini.

Voir Defining a view for the combo box selector of a foreign key [p 565] dans le Guide du dveloppeur.

UI bean

> **Attention**
>
> Depuis la version 5.8.0 de TIBCO EBX, il est recommand d'utiliser les widgets personnaliss la place des UI beans. Les widgets personnaliss proposent plus de fonctionnalits que les UI beans. Plus d'volutions seront apportes aux UI Beans. Voir widget [p 60] pour plus d'informations.

Cette proprit est disponible pour tous les types d'lments, except les tables.

Spcifie une classe Java permettant de personnaliser l'interface utilisateur associe cet lment dans un jeu de donnes. Un UI bean peut afficher l'lment diffremment et/ou modifier sa valeur en tendant la classe `UIBeanEditor`[API] dans l'API Java.

Transformation l'export

Cette proprit est disponible pour les champs et pour les groupes qui sont des noeuds terminaux.

Spcifie une classe Java dfinissant des oprations de transformation effectuer lors de la cration d'une archive partir d'un jeu de donnes associ. La transformation effectuer porte sur la valeur de l'lment courant.

Voir `NodeDataTransformer`[API] pour plus d'informations.

Proprits d'accs

Dfinit le mode d'accs pour l'lment courant, savoir:s'il peut tre crit et/ou lu.

- "Lecture & Ecriture" correspond au mode `RW` dans le XSD du modle de donnes.
- "Lecture seule" correspond au mode `R-` dans le XSD du modle de donnes.
- "Elment hors d'un jeu de donnes" correspond au mode `CC` dans le XSD du modle de donnes.
- "Noeud non terminal" correspond au mode `--` dans le XSD du modle de donnes.

Voir Access properties [p 561] dans le Guide du dveloppeur.

Catgorie du noeud

Dfinit les catgories permettant de restreindre la visualisation des donnes. Un noeud de catgorie "Hidden" est cach par dfaut dans un jeu de donnes.

	Restriction:la dfinition de catgorie ne s'applique pas aux noeuds d'enregistrement de tables, l'exception de la catgorie "Hidden". Voir Categories [p 567] dans le Guide du dveloppeur.
Mode de comparaison	Dfinit un mode de comparaison pour l'lment. Ce mode permet de restreindre la comparaison des donnes associes l'lment. • "Dfaut" implique que l'lment est pris en compte lors de la comparaison de donnes. • "Ignor" implique qu'aucune modification ne sera dtecte lors de la comparaison de deux entits modifies (jeux de donnes ou enregistrements). Lors de la fusion de donnes, les valeurs des lments ignors dans des jeux de donnes ou enregistrements modifis ne sont pas fusionnes. Les valeurs contenues dans de nouveaux jeux de donnes ou enregistrements sont fusionnes. Lors de l'import d'une archive, les valeurs des lments ignors dans des jeux de donnes ou enregistrements modifis ne sont pas importes. Les valeurs contenues dans de nouveaux jeux de donnes ou enregistrements sont importes. Voir Comparison mode [p 566] dans le Guide du dveloppeur.
Rgle d'application des dernires modifications	Indique si cet lment doit tre inclus ou non dans le service permettant d'appliquer d'autres enregistrements de la mme table les dernires modifications effectues. • "Dfaut" indique que la dernire modification effectue sur cet lment peut tre applique d'autres enregistrements. • "Ignor" indique que la dernire modification effectue sur cet lment ne peut pas tre applique d'autres enregistrements. Cet lment ne sera pas visible dans le service permettant d'appliquer les dernires modifications. Voir Apply last modifications policy [p 566] dans le Guide du dveloppeur.

Proprits avances des champs

Les proprits avances suivantes sont spcifiques aux champs.

Contrle de saisie sur valeur nulle

Implmente la proprit `osd:checkNullInput`. Cette proprit est utilise pour activer et vrifier une contrainte sur valeur `null` lors de la saisie utilisateur.

Par dfaut, pour permettre une saisie temporairement incomplte, la validation des lments obligatoires est effectue, non pas lors de la saisie utilisateur, mais pendant la validation du jeu de donnes. Si un lment obligatoire doit tre vrifi immdiatement lors de la saisie utilisateur, cette proprit doit avoir la valeur "oui".

> **Note**
>
> Une valeur est considre comme obligatoire si le "Nombre minimum de valeurs" est gal ou suprieur "1". Pour les lments terminaux, les valeurs obligatoires sont seulement vrifies dans les jeux de donnes activs. Pour les lments non terminaux, les valeurs sont vrifies indpendamment de l'tat d'activation du jeu de donnes.

Voir Constraints, triggers and functions [p 549] dans le Guide utilisateur.

Supprimer espaces

Supprimer espaces

Implmente la proprit osd:trim. Cette proprit permet d'indiquer si les espaces avant ou aprs une valeur doivent tre supprims lors de la saisie utilisateur. Si cette proprit n'est pas dfinie, alors les espaces avant ou aprs une valeur seront supprims par dfaut lors de la saisie utilisateur.

Voir Whitespace handling upon user input [p 550] dans le Guide utilisateur.

UI bean

Voir Proprits avances communes [p 61].

Fonction (valeur calcule)

Cette proprit est disponible pour les champs qui ne sont pas des cls primaires. Elle spcifie une classe Java permettant de calculer la valeur de ce champ programmatiquement. Peut tre utile si la valeur de ce champ dpend d'autres valeurs dans le rfrentiel, ou si le calcul de la valeur doit rcuprer des donnes depuis un systme externe.

Une fonction peut tre cre en implmentant l'interface `ValueFunction`[API].

Dsactiver validation

Indique si les contraintes dfinies sur le champ doivent tre dsactives. Cette proprit peut uniquement tre dfinie sur des champs calculs. Si "oui", les contraintes simples, avances et de cardinalit ne seront pas excutes lors de la validation des jeux de donnes associs au modle de donnes.

Transformation l'export

Voir Proprits avances communes [p 61].

Proprits d'accs

Voir Proprits avances communes [p 61].

Auto-incrment

Cette proprit est disponible uniquement pour les champs de type "entier" contenus dans une table. Si elle est active, la valeur du champ est calcule automatiquement lors de la cration d'un nouvel enregistrement. Peut tre utile pour les cls primaires, car l'auto-incrment gnre un identifiant unique pour chaque enregistrement. Deux attributs peuvent tre spcifis :

Valeur de dmarrage	Valeur initiale de l'auto-incrment. Si aucune valeur n'est dfinie, la valeur par dfaut est "1".
Pas de l'incrment	La valeur ajoute la valeur prcdente de l'auto-incrment. Si aucune valeur n'est dfinie, la valeur par dfaut est "1".
Dsactiver les contrles de l'auto-incrment	Indique s'il faut dsactiver le contrle de la valeur d'un champ auto-incrment par rapport la valeur maximale trouve dans la table en cours de mise jour.

Les valeurs auto-incrmentes ont le comportement suivant :

- Le calcul et l'allocation de la valeur de ce champ sont effectus ds qu'un nouvel enregistrement est insr et que la valeur du champ est indfinie.
- Aucune allocation ne peut tre ralise si l'insertion programmatique spcifie dj une valeur diffrente de `null`. Par consquent, cette allocation n'est pas effectue l'insertion d'un enregistrement en mode occultation ou rcriture.
- Si une archive importe spcifie cette valeur, alors elle est prserve.
- Une valeur nouvellement alloue est, autant que possible, unique au sein du rfrentiel.

  Plus prcisment, le caractre unique de l'allocation s'tend tous les jeux de donnes bass sur le modle de donnes, et galement tous les espaces de donnes. Ce dernier cas de figure permet de fusionner un espace de donnes son parent avec une garantie raisonnable d'absence de conflit lorsque la valeur auto-incrmente fait partie des enregistrements d'une cl primaire.

  Ce principe a une limitation trs spcifique:quand une opration majeure de mise jour spcifiant des valeurs est ralise en simultan d'une opration allouant une valeur au mme champ, il est possible que cette dernire opration alloue une valeur qui sera fixe par la premire opration (il n'y a pas de verrouillage entre plusieurs espaces de donnes).

Voir Auto-incremented values [p 553] dans le Guide du dveloppeur.

Affichage par dfaut

Voir Proprits avances communes [p 59].

Catgorie du noeud

Voir Proprits avances communes [p 61].

Champ hrit

Dfinit une relation entre le champ courant et un champ dans une autre table afin de chercher sa valeur automatiquement.

Enregistrement source	Une cl trangre ou une squence de cls trangres, spares par des espaces, permettant de trouver l'enregistrement dont hrite le champ courant. Si cette proprit n'est pas renseigne, alors l'lment courant sera utilis comme source d'hritage de champ.
lment source	Chemin XPath dfinissant l'lment hriter. L'lment source doit tre terminal, se trouver dans "Enregistrement source", et son type doit tre le mme que le type de ce champ. Cette proprit est obligatoire pour l'hritage de champ.

Voir hritage [p 27] dans le glossaire.

Pour plus d'informations, voir aussi Champs hrits [p 296].

Proprits avances des tables

Les proprits avances suivantes sont spcifiques aux tables.

Table

Cl primaire	Liste des champs composant la cl primaire de la table. Il est possible d'ajouter ou de supprimer des champs de la cl primaire. Chaque champ de la cl est identifi par un chemin XPath absolu qui dbute sous la table. Si la cl est composite, la liste des champs doit tre spare par des espaces. Par exemple, "/name /startDate".
Prsentation	Spcifie la manire dont les enregistrements de cette table sont affichs dans l'interface utilisateur dans les jeux de donnes associs.
Labellisation des enregistrements	Dfinit les champs composant le libell par dfaut et les libells localiss pour les enregistrements de la table. Peut aussi dfinir une classe Java pour affecter le libell programmatiquement, ou dfinir le libell dans une hirarchie. Cette classe Java doit implmenter l'interface `UILabelRenderer`[API] ou `UILabelRendererForHierarchy`[API] de l'API Java. **Attention:**Les droits d'accs dfinis dans les jeux de donnes associs ne sont pas appliqus lors de l'affichage des libells. Les champs habituellement cachs cause de droits d'accs seront affichs dans les libells des enregistrements.
Prsentation par dfaut des groupes dans un formulaire	Spcifie la politique d'affichage par dfaut des groupes contenus dans cette table. Si cette proprit n'est pas dfinie, alors la politique par dfaut sera utilise pour afficher les groupes. Voir Record form: rendering mode for nodes [p 414] dans le Guide d'administration. **Prsentation active des groupes** : spcifie un mode de prsentation autoris en complment de "Dvelopp" et "Rduit", qui sont toujours disponibles. Les liens doivent tre activs sur la table afin de dfinir spcifiquement le mode de prsentation des groupes en tant que liens. **Prsentation par dfaut des groupes**:spcifie la prsentation par dfaut des groupes contenus dans cette table. Si un groupe ne spcifie aucune vue par dfaut, alors le mode de prsentation par dfaut dfini par cette table sera utilis. En fonction des performances du rseau et du navigateur, ajustez la manire d'afficher chaque groupe du formulaire. En ce qui concerne le poids de la page tlcharge, le mode "Lien" est lger, les modes "Dvelopp" et "Rduit" sont plus lourds.

	Note : quand les onglets sont activs dans une table, tous les groupes qui utiliseraient les liens dans les autres cas sont convertis automatiquement en mode "Rduit". Cela permet d'viter les complexits d'affichage qui se posent quand les liens et les onglets sont dans la mme interface utilisateur.
Prsentation spcifique d'un formulaire	Dfinit une prsentation spcifique pour personnaliser le formulaire d'un enregistrement dans les jeux de donnes associs. Voir `UIForm`[API] et `UserServiceRecordFormFactory`[API] pour plus d'informations.
Barres d'outils	Dfinit les barres d'outils afficher dans la table. Les barres d'outils peuvent tre consultes et modifies dans la section *Configuration > Barre d'outils*. **Entte de table:** Dfinit la barre d'outils afficher en entte de la vue table par dfaut. **Ligne de table:** Dfinit la barre d'outils afficher pour chaque ligne de la vue table par dfaut. **Entte de formulaire:** Dfinit la barre d'outils afficher en entte du formulaire d'un enregistrement. **Entte de hirarchie:** Dfinit la barre d'outils afficher en entte de la hirarchie par dfaut de la table. Voir Barre d'outils [p 79] pour plus d'informations.
Historisation	Spcifie si l'historisation est active, et le niveau de garantie demand. Les profils d'historisation peuvent tre modifis dans la section **Administration > Historique et journaux**. Voir Configuration de l'historique dans le rfrentiel [p 273] pour plus d'informations.
Index	Proprit permettant d'indexer certains champs de la table. L'indexation amliore le temps d'accs des requtes portant sur ces champs. La combinaison des champs dans chaque index doit tre unique. **Nom de l'index**: Nom unique pour cet index. **Champs indexer**: Les champs indexer, o chaque champ d'index est identifi par un chemin XPath absolu qui dbute sous la table.
Filtres spcifiques	Dfinit des filtres pour afficher uniquement les enregistrements correspondant certains critres.
Actions	Indique les actions autorises ou interdites dans le contexte d'un jeu de donnes. Toutes les actions sont autorises par

dfaut, cependant des droits d'accs peuvent tre dfinis dans les jeux de donnes associs pour restreindre ces actions.

Contraintes d'unicit

Indique les champs dont les valeurs doivent tre uniques dans la table.

Triggers la mise jour

Spcifie des classes Java dfinissant des mthodes excutes automatiquement lorsque des oprations sont effectues:cration, mise jour, suppression, etc.

Un trigger natif est inclus par dfaut pour lancer les workflows de donnes.

Voir Triggers [p 552] dans le Guide du dveloppeur.

Proprits d'accs

Voir Proprits avances communes [p 61].

Affichage par dfaut

Voir Proprits avances communes [p 59].

Catgorie du noeud

Voir Proprits avances communes [p 61].

Proprits avances des groupes

Les proprits avances suivantes sont spcifiques aux groupes.

Classe du conteneur de valeurs (JavaBean)

Dfinit une classe Java pour contenir les valeurs des lments situs sous le groupe. La classe Java spcifie doit tre conforme la spcification JavaBean. Cela signifie que chaque lment enfant de ce groupe doit correspondre une proprit de la classe Java. Toutes les proprits de la classe Java doivent avoir des mthodes d'accs (getters et setters).

UI bean

Voir Proprits avances communes [p 61].

Transformation l'export

Voir Proprits avances communes [p 61].

Proprits d'accs

Voir Proprits avances communes [p 61].

Affichage par dfaut

Visibilit	Voir Proprits avances communes [p 59].
Prsentation	Dfinit la manire dont ce groupe sera prsent dans un jeu de donnes. Si cette proprit n'est pas dfinie, alors la vue par dfaut dfinie par la table parente sera utilise. Les options "Onglet" et "Lien" sont disponibles uniquement si la table parente a activ ces options de prsentation. ***Proprits de l'onglet*** **Position:** Cet attribut spcifie la position de l'onglet par rapport tous les onglets dfinis dans le modle. Cette position est utilise pour ordonnancer la liste des onglets au moment de l'affichage d'un formulaire. Si cette proprit n'est pas dfinie, alors l'onglet sera positionn en fonction de la position du groupe dans le modle de donnes.

Catgorie du noeud

Voir Proprits avances communes [p 61].

Concepts apparentés*Contrles sur les lments du modle de donnes [p 71]*

CHAPITRE 9

Contrles sur les lments du modle de donnes

Aprs la cration d'un lment, compltez sa dfinition avec des contrles supplmentaires.

Voir aussi*Proprits des lments du modle de donnes* *[p 55]*

Ce chapitre contient les sections suivantes :

1. Contrles simples
2. Contrles avancs

9.1 Contrles simples

Il est possible d'tablir des contrles simples sur le contenu d'un champ en dfinissant des contrles statiques. Les contrles disponibles dpendent du type du champ.

Longueur fixe	Nombre exact de caractres requis pour ce champ.
Longueur minimale	Nombre minimum de caractres requis pour ce champ.
Longueur maximale	Nombre maximum de caractres autoriss pour ce champ.
Pattern	Expression rgulire laquelle la valeur du champ doit tre conforme. Il est interdit de dfinir un pattern la fois sur un champ et sur son type de donne.
Partie dcimale	Nombre maximum de chiffres autoriss dans la partie dcimale de la valeur d'un champ de type dcimal.
Nombre maximum de chiffres	Nombre maximum de chiffres composant la valeur d'un champ de type entier ou dcimal.
numration	Dfinit une liste de valeurs possibles pour ce champ. Si une numration est dfinie la fois sur ce champ et sur son type de donnes, alors une numration reprsentant l'intersection entre ces deux numrations sera utilise dans les jeux de donnes associs au modle de donnes.
Suprieur [constante]	Dfinit la valeur minimale autorise pour ce champ.
Infrieur [constante]	Dfinit la valeur maximale autorise pour ce champ.

Voir Facettes XML schema supportes [p 537].

9.2 Contrles avancs

Il est possible d'tablir des contrles avancs sur le contenu d'un lment en dfinissant des contrles dynamiques contextuels. Les contrles disponibles pour un lment dpendent de sa nature (table, groupe, etc.) et de son type de donnes.

Voir aussi*Dynamic constraints [p 540]*

Contrainte de cl trangre	
Table	Dfinit la table cible par la cl trangre. Une cl trangre rfrence une table dans le mme jeu de donnes par dfaut. Elle peut aussi rfrencer une table d'un autre jeu de donnes du mme espace de donnes, ou un jeu de donnes d'un autre espace de donnes.
Mode	Emplacement de la table cible par la cl trangre. "Dfaut":modle courant. "Autre jeu de donnes":jeu de donnes qui se trouve dans le mme espace de donnes. "Autre espace de donnes":jeu de donnes appartenant un espace de donnes diffrent.
Table rfrence	Expression XPath indiquant l'emplacement de la table. Par exemple, `/racine/MaTable`.
Jeu de donnes rfrenc	Obligatoire si la table est dans un autre jeu de donnes. Le nom unique du jeu de donnes contenant la table rfrence.
Espace de donnes rfrenc	Obligatoire si la table est dans un autre espace de donnes. Nom unique de l'espace de donnes contenant la table rfrence.
Libell	Dfinit les champs composant un libell par dfaut et des libells localiss pour prsenter les enregistrements de la table cible. Peut aussi spcifier une classe Java qui dfinit le libell programmatiquement quand "Expression XPath" a la valeur "Non". Cette classe Java doit implmenter l'interface `TableRefDisplay`[API] de l'API Java. **Attention:**Les droits d'accs dfinis dans les jeux de donnes associs ne sont pas appliqus lors de l'affichage des libells. Les champs habituellement cachs cause de droits d'accs seront affichs dans les libells des enregistrements.
Filtre	Dfinit un filtre de cl trangre en utilisant une expression XPath. Peut aussi spcifier une classe Java qui implmente l'interface `TableRefFilter`[API] de l'API Java.

Suprieur [variable]	Dfinit un champ qui spcifie la valeur minimale autorise pour ce champ.
Infrieur [variable]	Dfinit un champ qui spcifie la valeur maximale autorise pour ce champ.
Longueur fixe [dynamique]	Dfinit un champ qui spcifie le nombre exact de caractres requis pour ce champ.
Longueur minimale [dynamique]	Dfinit un champ qui spcifie le nombre minimum de caractres requis.
Longueur maximale [dynamique]	Dfinit un champ qui spcifie le nombre maximum de caractres autoriss.
Valeurs exclues	Dfinit une liste de valeurs non autorises pour ce champ.
Plage exclue de valeurs	Dfinit une plage de valeurs non autorises pour ce champ. **Valeur maximale exclue :** La valeur maximale non autorise pour ce champ. **Valeur minimale exclue :** La valeur minimale non autorise pour ce champ.
Contrainte spcifique (composant)	Spcifie une ou plusieurs classes Java qui implmentent l'interface `Constraint`[API] de l'API Java. Voir Programmatic constraints [p 544] pour plus d'informations.
Enumration spcifique (composant)	Spcifie une classe Java pour dfinir une numration. La classe doit dfinir une liste ordonne de valeurs, en implmentant l'interface `ConstraintEnumeration`[API] de l'API Java.
Enumration alimente par un autre noeud	Spcifie un champ qui dfinit les valeurs autorises pour cette numration. Le champ spcifi doit tre une liste ou doit dfinir une numration.
Configuration des espaces de donnes	Dfinit quels sont les espaces de donnes pouvant tre rfrencs par un champ de type Identifiant d'espaces de donnes (osd:dataspaceKey). • Inclusions Dfinit les espaces de donnes qui peuvent tre rfrencs par ce champ. **Pattern:**Dfinit une expression rgulire laquelle le nom d'un espace de donnes doit tre conforme.

Type: Dfinit le type d'espaces de donnes (branches ou versions) qui peuvent tre rfrencs par ce champ. Si non dfinie, cette restriction s'appliquera aux espaces de donnes de type branche.

Inclure les descendants: Dfinit si les espaces de donnes enfants ou descendants sont inclus dans cet ensemble. Si non dfinie, cette restriction ne s'appliquera pas aux espaces de donnes enfants. Si "Aucun" alors les enfants et descendants des espaces de donnes qui vrifient le pattern dfini ne sont pas inclus. Si "Tous les descendants" alors toutes les branches et versions descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Toutes les branches descendantes" alors toutes les branches descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Toutes les versions descendantes" alors toutes les versions descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Branches enfants" alors seules les branches directes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si l'espace de donnes qui vrifie le pattern est une version, alors les branches qui sont des enfants directs de cette version sont incluses. Si l'espace de donnes qui vrifie le pattern est une branche, alors sont incluses les branches qui sont des enfants directs des versions qui sont des enfants directs de cette branche. Si "Versions enfants" alors seules les versions directes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si l'espace de donnes qui vrifie le pattern est une branche, alors les versions qui sont des enfants directs de cette branche sont incluses. Si l'espace de donnes qui vrifie le pattern est une version, alors sont incluses les versions qui sont des enfants directs des branches qui sont des enfants directs de cette version.

- Exclusions

  Dfinit les espaces de donnes qui ne peuvent pas tre rfrencs par ce champ. Les exclusions sont ignores si la configuration ne dfinit pas d'inclusion.

  Pattern: Dfinit une expression rgulire laquelle le nom d'un espace de donnes doit tre conforme.

  Type: Dfinit le type d'espaces de donnes (branches ou versions) qui peuvent tre rfrencs par ce champ. Si non dfinie, cette restriction s'appliquera aux espaces de donnes de type branche.

  Inclure les descendants: Dfinit si les espaces de donnes enfants ou descendants sont inclus dans cet ensemble. Si non dfinie, cette restriction ne s'appliquera pas aux espaces de donnes enfants. Si "Aucun" alors

les enfants et descendants des espaces de donnes qui vrifient le pattern dfini ne sont pas inclus. Si "Tous les descendants" alors toutes les branches et versions descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Toutes les branches descendantes" alors toutes les branches descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Toutes les versions descendantes" alors toutes les versions descendantes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si "Branches enfants" alors seules les branches directes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si l'espace de donnes qui vrifie le pattern est une version, alors les branches qui sont des enfants directs de cette version sont incluses. Si l'espace de donnes qui vrifie le pattern est une branche, alors sont incluses les branches qui sont des enfants directs des versions qui sont des enfants directs de cette branche. Si "Versions enfants" alors seules les versions directes des espaces de donnes qui vrifient le pattern dfini sont incluses. Si l'espace de donnes qui vrifie le pattern est une branche, alors les versions qui sont des enfants directs de cette branche sont incluses. Si l'espace de donnes qui vrifie le pattern est une version, alors sont incluses les versions qui sont des enfants directs des branches qui sont des enfants directs de cette version.

- Filtre

  Dfinit un filtre pour accepter ou refuser des espaces de donnes dans le contexte d'un jeu de donnes ou d'un enregistrement. Ce filtre est uniquement utilis par le composant de saisie ddi ce type de champ. Ce filtre n'est pas utilis lors de la validation de ce champ. Des contrles additionnels peuvent tre dfinis en utilisant une contrainte spcifique (composant). Un filtre est dfini par une classe Java qui implmente l'interface `DataspaceSetFilter`[API].

Configuration des jeux de donnes

Dfinit quels sont les jeux de donnes pouvant tre rfrencs par un champ de type Identifiant de jeux de donnes (osd:datasetName).

- Inclusions

  Dfinit les jeux de donnes qui peuvent tre rfrencs par ce champ.

  Pattern: Dfinit une expression rgulire laquelle le nom d'un jeu de donnes doit tre conforme.

  Inclure les descendants: Dfinit si les jeux de donnes enfants ou descendants sont inclus dans cet ensemble.

Si non dfinie, cette restriction ne s'appliquera pas aux jeux de donnes enfants.

- Exclusions

  Dfinit les jeux de donnes qui ne peuvent pas tre rfrencs par ce champ. Les exclusions sont ignores si la configuration ne dfinit pas d'inclusion.

  Pattern: Dfinit une expression rgulire laquelle le nom d'un jeu de donnes doit tre conforme.

  Inclure les descendants: Dfinit si les jeux de donnes enfants ou descendants sont inclus dans cet ensemble. Si non dfinie, cette restriction ne s'appliquera pas aux jeux de donnes enfants.

- Filtre

  Dfinit un filtre pour accepter ou refuser des jeux de donnes dans le contexte d'un jeu de donnes ou d'un enregistrement. Ce filtre est uniquement utilis par le composant de saisie ddi ce type de champ. Ce filtre n'est pas utilis lors de la validation de ce champ. Des contrles additionnels peuvent tre dfinis en utilisant une contrainte spcifique (composant). Un filtre est dfini par une classe Java qui implmente l'interface `DatasetSetFilter`[API].

Proprits de validation

Chaque contrainte, except celles utilisant une classe Java, peut dfinir des messages de validation. Il est possible d'associer une svrit ces messages de validation et ils peuvent tre localiss en utilisant les proprits suivantes :

Validation	Spcifie le message de validation et la svrit associe la contrainte.
Svrit	Spcifie la svrit de la contrainte. Les valeurs possibles sont : "Erreur", "Avertissement", et "Information".
Gestion des erreurs	Spcifie le comportement de la contrainte en cas d'erreur de validation. Il est possible d'indiquer que la contrainte doit toujours tre valide pour toute opration (mise jour d'un jeu de donnes, cration, mise jour et suppression d'un enregistrement), ou lorsqu'un utilisateur soumet un formulaire. Dans ce cas, toutes les saisies ou oprations qui rendraient la contrainte invalide seront rejetes et les donnes seront non modifies. Si cette proprit n'est pas spcifie, alors la contrainte bloquera uniquement les erreurs lors de la soumission d'un formulaire, sauf dans le cas de modles relationnels o les contraintes de cls trangres bloquent par dfaut toutes les erreurs. Cette proprit est uniquement disponible sur les contrles statiques, valeurs exclues, plage exclue de valeurs et contraintes de cls trangres. Le caractre bloquant pour toutes les oprations ne s'applique pas sur les filtres de cl trangre. Une contrainte de cl trangre n'est donc pas bloquante si un enregistrement rfrenc existe mais ne rpond pas aux critres de filtrage. Il n'est pas possible de dfinir cette proprit sur les contraintes structurelles dfinies dans les modles relationnels. Si cette proprit est dfinie sur les contraintes "Longueur fixe", "Longueur maximum", "Nombre total de chiffres" et "Partie dcimale" dans des modles relationnels, une erreur sera leve lors de la compilation du modle. Le caractre bloquant d'une contrainte est ignor lors de l'import d'archives. Toutes les contraintes bloquantes, sauf les contraintes dites structurelles, sont dsactives lors de l'import d'archives.
Message	Message afficher lorsque la valeur de ce champ dans un jeu de donnes ne respecte pas cette contrainte. Ce message peut tre localis.

Concepts apparentés*Proprits des lments du modle de donnes* *[p 55]*

CHAPITRE **10**

Barres d'outils

Ce chapitre contient les sections suivantes :

1. Dfinition

10.1 Dfinition

Une barre d'outils permet de personnaliser les boutons et menus affichs lors de la consultation de tables ou d'enregistrements dans un jeu de donnes. Les barres d'outils sont configurables dans le modle de donnes via la section 'Configuration'.

Ajoutez une barre d'outils partir de la section *Barres d'outils* du panneau de navigation, en cliquant sur la flche situe droite de l'lment *[Tous les lments]*, puis en slectionnant l'option *Crer barre d'outils*. Suivez ensuite l'assistant de cration pour crer une barre d'outils. Une barre d'outils dfinit les informations suivantes:

Nom	Nom de la barre d'outils. Le nom de la barre d'outils doit tre unique dans le modle de donnes, il n'est pas possible de crer plusieurs barres d'outils avec le mme nom. Les tables et associations pouvant utiliser des barres d'outils, utilisent ce nom pour dfinir la barre d'outils utiliser.
Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.
Modle de barre d'outils	Permet de crer une barre d'outils partir d'une barre d'outils par dfaut.
Emplacements	Dfinit les emplacements pour lesquels la barre d'outils peut tre utilise dans les jeux de donnes associs.

Dfinir la structure d'une barre d'outils

Une barre d'outils peut dfinir les lments suivants :

- Bouton action [p 81]
- Bouton menu [p 82]

- Sparateur [p 82]
- Groupe de menu [p 83]
- Action de menu [p 84]
- Sous-menu [p 84]

Ajoutez un de ces lments sous une barre d'outils ou un lment existant en cliquant sur la flche situe droite de l'lment existant, puis en slectionnant une option de cration parmi celles prsentes dans le menu. Suivez ensuite l'assistant de cration pour crer un lment.

Bouton action

Ce type d'lment permet d'associer une action un bouton dans une barre d'outils. L'action est excute lorsque l'utilisateur clique sur le bouton associ dans une barre d'outils. Un lment de type *Bouton action* dfinit les informations suivantes:

Cible	Dfinit si le service s'excute sur la slection courante ou dans un composant web. Un service s'excutant dans un composant web n'a pas accs la slection courante, mais il est possible de spcifier une slection et des paramtres spcifiques.
Service	Dfinit le service excuter lorsque l'utilisateur clique sur le bouton. Il est possible de slectionner un service fourni, ou un service utilisateur dfini dans un module ou dans le modle de donnes courant. Si la cible 'Composant web' est slectionne, le service devra avoir t dclar comme utilisable en tant que composant web dans les barres d'outils. **Voir aussi** *`WebComponentDeclarationContext.setAvailableAsToolbarAction`*[API]
Taille de la modale	Taille de la fentre modale affichant le composant web.
Redimensionnable	Indique si la fentre modale peut tre redimensionne.
Paramtres	Paramtres du service. Les paramtres ne peuvent tre spcifis que si la cible du service est 'Composant web'.
Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.
Rendu	Dfinit la manire d'afficher cet lment dans les jeux de donnes utilisant la barre d'outils. Il est possible d'afficher : l'icne seule, le texte seul, le texte avec l'icne sur la gauche ou le texte avec l'icne sur la droite.
Icne	Icne afficher. Il est possible d'utiliser une icne parmi un ensemble d'icnes proposes, ou de faire rfrence une icne en utilisant une URL.
Est en surbrillance	Indique si le bouton doit tre par dfaut en surbrillance.

> **Note**
>
> Un lment de type *Bouton action* peut uniquement tre cr sous un lment de type *barre d'outils*.

Bouton menu

Ce type d'lment permet de dfinir un menu qui est affich lorsque l'utilisateur clique sur le bouton associ dans une barre d'outils. Un lment de type *Bouton menu* dfinit les informations suivantes:

Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.
Rendu	Dfinit la manire d'afficher cet lment dans les jeux de donnes utilisant la barre d'outils. Il est possible d'afficher : l'icne seule, le texte seul, le texte avec l'icne sur la gauche ou le texte avec l'icne sur la droite.
Icne	Icne afficher. Il est possible d'utiliser une icne parmi un ensemble d'icnes proposes, ou de faire rfrence une icne en utilisant une URL.
Est en surbrillance	Indique si le bouton doit tre par dfaut en surbrillance.

> **Note**
>
> Un lment de type *Bouton menu* peut uniquement tre cr sous un lment de type *barre d'outils*.

Sparateur

Ce type d'lment permet d'insrer un sparateur sous la forme d'espacements entre deux lments d'une barre d'outils.

> **Note**
>
> Un lment de type *Sparateur* peut uniquement tre cr sous un lment de type *barre d'outils*.

Groupe de menu

Ce type d'lment permet de dfinir un groupe d'lments dans un menu. Un lment de type *Groupe de menu* dfinit les informations suivantes:

Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.
Type de groupe	Indique le type de groupe de menu crer : - 'Local' permet de crer un groupe de menu vide. - 'Groupe de services' permet d'assigner un groupe de services existant ce groupe de menu. - 'Menu builder' permet d'assigner un menu prdfini ce groupe de menu. Une fois cr, il n'est pas possible de changer le type de ce groupe de menu.
Nom du groupe de service	Dfinit un groupe de services existant rutiliser. Un groupe de services est dclar dans un module et peut inclure d'autres groupes de services. Tous les services contenus dans ce groupe seront affichs dans les jeux de donnes associs.
Nom du Menu builder	Indique le menu prdfini assigner ce groupe de menu: - 'Menu par dfaut "Actions"' correspond au menu 'Actions' de la barre d'outil par dfaut. Les services standards et utilisateurs y sont prsents sans distinction. - 'Menu par dfaut "Actions"' (avec sparateur)' a le mme contenu que le menu 'Actions' de la barre d'outil par dfaut, mais les services standards et utilisateurs y sont prsents sparment (d'abord les services standards, puis les services utilisateurs).
Services exclus	Dfinit les services exclure du groupe de services rutiliser. Les services exclus ne seront pas affichs dans les jeux de donnes associs.
Groupes de services exclus	Dfinit les groupes exclure du groupe de services rutiliser. Les services contenus dans les groupes exclure ne seront pas affichs dans les jeux de donnes associs.
Politique d'affichage	Si "Filtrage intelligent", les services configurs en accs direct, c'est dire via un bouton action ou une action de menu, seront retirs de la gnration automatique de ce groupe.

> **Note**
>
> Un lment de type *Groupe de menu* peut uniquement tre cr sous les lments suivants:

- Bouton menu
- Sous-menu

Action de menu

Ce type d'lment permet d'associer une action une entre d'un menu dans une barre d'outils. L'action est excute lorsque l'utilisateur clique sur l'entre correspondante dans un menu. Un lment de type *Action de menu* dfinit les informations suivantes:

Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.
Cible	Dfinit si le service s'excute sur la slection courante ou dans un composant web. Un service s'excutant dans un composant web n'a pas accs la slection courante, mais il est possible de spcifier une slection et des paramtres spcifiques.
Service	Dfinit le service excuter lorsque l'utilisateur clique sur le bouton. Il est possible de slectionner un service fourni, ou un service utilisateur dfini dans un module ou dans le modle de donnes courant. Si la cible 'Composant web' est slectionne, le service devra avoir t dclar comme utilisable en tant que composant web dans les barres d'outils. **Voir aussi** *WebComponentDeclarationContext.setAvailableAsToolbarAction*[API]
Taille de la modale	Taille de la fentre modale affichant le composant web.
Redimensionnable	Indique si la fentre modale peut tre redimensionne.
Paramtres	Paramtres du service. Les paramtres ne peuvent tre spcifis que si la cible du service est 'Composant web'.

> **Note**
>
> Un lment de type *Action de menu* peut uniquement tre cr sous un lment de type *Groupe de menu*.

Sous-menu

Ce type d'lment permet d'ajouter un sous-menu dans un menu d'une barre d'outils. Un *Sous-menu* dfinit les informations suivantes:

Libell et description	Libells et descriptions internationaliss affichs l'utilisateur.

> **Note**
>
> Un lment de type *Sous-menu* peut uniquement tre cr sous un lment de type Groupe de menu.

Supprimer des lments

Tous les lments d'une barre d'outils peuvent tre supprims de celle-ci en utilisant la flche situe droite de l'lment supprimer.

Si un lment contenant d'autres lments est supprim, alors la suppression est effectue rcursivement sur tous les lments situs sous l'lment supprim.

Dupliquer des lments existants

Pour dupliquer un lment, cliquez sur la flche situe droite de l'lment dupliquer. Spcifiez le nom et les proprits de l'lment dupliqu. Toutes les proprits de l'lment source sont dupliques.

L'lment dupliqu est ajout au mme niveau que l'lment d'origine, en dernire position. Lorsqu'un lment contenant d'autres lments est dupliqu, tous les sous-lments sont dupliqus avec leurs proprits.

Dplacer des lments

Pour dplacer un lment, cliquez sur la flche et slectionnez l'option de dplacement souhaite.

Associer avec des tables existantes

Pour associer une barre d'outils avec des tables existantes, cliquez sur la flche et slectionnez l'option *Associer des tables*. Ce service permet de dfinir une barre d'outils en tant que barre d'outils par dfaut d'un ensemble de tables. Pour cela, dfinissez les positions cibles de la barre d'outils et slectionnez les tables ou types de donnes complexes, dfinissant les proprits de la table, associer la barre d'outils. La barre d'outils sera dfinie par dfaut aux positions dfinies sur les tables et types slectionnes.

Exporter les barres d'outils

Il est possible d'exporter les barres d'outils dfinies dans le modle dans un document XML. Pour cela, slectionner l'option *Export XML* disponible dans le menu *Actions* de la section 'Barres d'outils'. Suivez ensuite l'assistant pour exporter les barres d'outils.

> **Note**
>
> Une slection de barres d'outils peut tre exporte en slectionnant dans la section 'Barres d'outils' les barres d'outils exporter, puis en slectionnant l'option *Export XML* disponible dans le menu *Actions*. Les barres d'outils peuvent aussi tre exportes en utilisant le service d'export des modles de donnes accessible depuis le menu **'Actions' du modle de donnes** [p 35] dans le panneau de navigation.

Voir aussi*Import et export de fichiers XML Schema Document (XSD)* *[p 87]*

Importer des barres d'outils

Il est possible d'importer des barres d'outils existantes partir d'un document XML. Pour cela, slectionner l'option *Import XML* disponible dans le menu *Actions* de la section Barres d'outils. Suivez ensuite l'assistant pour importer les barres d'outils.

> **Note**
>
> Les barres d'outils peuvent aussi tre importes en utilisant le service d'import de modles de donnes accessible depuis le menu **'Actions' du modle de donnes** [p 35] dans le panneau de navigation.

Voir aussi*Import et export de fichiers XML Schema Document (XSD)* *[p 87]*

Voir aussi *Utilisation des barres d'outils* *[p 67]*

CHAPITRE 11

Actions sur les modles de donnes existants

Lorsque le modle de donnes est cr, des actions sont disponibles dans le menu **"Actions" du modle de donnes** [p 35] dans le panneau de navigation.

Ce chapitre contient les sections suivantes :

1. Validation du modle de donnes
2. Import et export de fichiers XML Schema Document (XSD)
3. Duplication d'un modle de donnes
4. Suppression d'un modle de donnes

11.1 Validation du modle de donnes

Il est possible de valider un modle de donnes en slectionnant **"Actions" du modle de donnes** > **Valider** dans le panneau de navigation. Les ventuels messages issus de la validation du modle de donnes sont prsents dans un rapport. Depuis le rapport de validation, cliquez sur le bouton **Revalider** pour mettre jour ce rapport, ou cliquez sur le bouton **Rinitialiser le rapport de validation** pour supprimer tous les messages de validation actuellement associs au modle de donnes afin de pouvoir relancer une validation complte.

Voir Validation [p 324] pour obtenir plus d'informations concernant la validation incrmentale de donnes.

11.2 Import et export de fichiers XML Schema Document (XSD)

TIBCO EBX fournit des services intgrs pour importer et exporter des fichiers XML Schema Document (XSD). Les services d'import et d'export sont accessibles depuis le menu **"Actions" du modle de donnes** [p 35] dans le panneau de navigation. Un import ou export est toujours effectu sur l'intgralit du modle de donnes. Lors d'un import, la structure du modle de donnes courant est entirement remplace par le contenu du document XML Schema import. Lors d'un export, le modle de donnes complet est export dans le document XML Schema cible.

Pour importer un fichier XSD, le fichier doit tre valide et conforme aux rgles de validation du rfrentiel EBX. Si le document dclare des ressources situes dans un module, le module doit tre dclar aussi dans

la configuration du modle de donnes. Si le module n'a pas t dclar, le fichier XSD ne pourra pas tre import. Voir Proprits du modle de donnes [p 43] pour plus d'informations sur la dclaration des modules.

Pour importer un modle, slectionnez *Importer XSD* dans le menu **Actions** situ dans le panneau de navigation du modle de donnes.

Un document XML Schema (XSD) peut tre import partir du systme de fichier local. Pour cela, slectionnez *Importer partir d'un document local*:

- **Modle de donnes :** chemin du document XSD importer dans le systme de fichiers local.
- **Barres d'outils :** chemin du document XML importer dans le systme de fichiers local qui contient les barres d'outils utilises par le modle de donnes.

Il est possible d'importer un document XML Schema (XSD) contenu dans le rfrentiel. Pour cela, slectionnez *Importer partir du rfrentiel de modles* :

- **Modle :** nom du modle de donnes importer.

Un modle de donnes XSD peut aussi tre import dans un module. L'import d'un modle de donnes XSD depuis un module utilise les proprits suivantes :

Module	Module qui contient le modle de donnes.
Chemin du module	Localisation physique du module sur le systme de fichier du serveur.
Chemin des sources	Codes sources utiliss pour configurer les "Rgles et objets mtier". Si le chemin est relatif, il sera rsolu partir du "Chemin du module". Cette proprit est obligatoire si le modle dfinit des lments programmatiques.
Modle	Modle de donnes importer.

> **Note**
>
> Les fichiers XSD et XML importer doivent tre encods en "UTF-8". Les fichiers XSD et XML exports sont toujours encods en "UTF-8".

11.3 Duplication d'un modle de donnes

Pour dupliquer un modle de donnes, slectionnez "Dupliquer" dans le menu **Actions** du modle. Nommez le nouveau modle de donnes. Ce nom doit tre unique dans le rfrentiel.

11.4 Suppression d'un modle de donnes

Pour supprimer un modle de donnes, slectionnez "Supprimer" dans le menu **Actions** du modle de donnes [p 35]. Quand un modle de donnes est supprim, toutes les publications associes au modle restent disponibles. Si un nouveau modle de donnes est recr avec le mme nom que celui qui a t supprim,

le nouveau modle de donnes sera rassoci avec toutes les publications existant dans le rfrentiel. Au moment de la publication, il sera possible de confirmer le remplacement d'une publication existante.

> **Note**
>
> Seul un administrateur peut supprimer les publications d'un modle de donnes dans la section "Administration".

Voir Publication des modles de donnes [p 91] pour plus d'informations sur le processus de publication.

CHAPITRE **12**

Publication du modle de donnes

Ce chapitre contient les sections suivantes :

1. A propos des publications
2. Modes de publication
3. Mode de publication "Embarqu"

12.1 A propos des publications

Chaque jeu de donnes dans le rfrentiel TIBCO EBX bas sur un **modle de donnes embarqu** est associ une publication d'un modle de donnes, et non directement au modle de donnes dfini dans le **Data Model Assistant**. Une publication est cre la premire fois qu'un modle de donnes est publi en utilisant le bouton **Publier** du panneau de navigation. Il est possible, partir d'une publication, de crer des jeux de donnes dont la structure sera base sur la structure du modle de donnes publi.

> **Note**
>
> Le bouton **Publier** est uniquement affich pour les utilisateurs qui ont le droit de publier le modle de donnes. Voir Permissions du modle de donnes [p 41] pour plus d'informations.

Les jeux de donnes tant bass sur des publications, les modifications faites sur le modle de donnes n'impacteront pas les jeux de donnes existants. Les jeux de donnes seront impacts uniquement lors de la mise jour d'une publication existante.

12.2 Modes de publication

Un modle de donnes peut tre publi en mode "Embarqu" ou "Dans un module". Le mode de publication "Embarqu" gnre une publication gre et persiste dans le rfrentiel EBX et possde des fonctionnalits spcifiques ddies la gestion de version. Le mode de publication "Dans un module" cre un fichier XML Schema Document (XSD) l'intrieur d'un module qui n'est pas gr par le rfrentiel EBX.

Selon la configuration du modle de donnes, EBX dtermine automatiquement le processus de publication utiliser lorsque l'on clique sur le bouton **Publier** dans le panneau de navigation. Quand un modle de donnes spcifie le mode de publication "Dans un module" ainsi qu'un fichier XSD cibler, le processus de publication gnre le fichier XSD dans le module dfini dans la configuration.

12.3 Mode de publication "Embarqu"

La premire fois qu'un modle de donnes est publi en mode embarqu, une nouvelle publication avec le mme nom que le modle est automatiquement cre dans le rfrentiel. Si diffrentes publications ont t cres partir du modle, slectionnez celle mettre jour.

Voir Consultation et cration des publications [p 92] pour plus d'informations sur l'utilisation de diffrentes publications.

Pendant le processus de publication, si le modle de donnes a dj t publi, il est possible de visualiser dans une interface de comparaison les diffrences structurelles introduites par la nouvelle publication.

Le processus de publication permet aussi de crer une image en lecture seule de l'tat actuel du modle de donnes. Cette image sera utile si un prcdent tat du modle doit tre restaur aprs plusieurs modifications et publications du modle de donnes.

> **Note**
>
> Les images sont des archives statiques du modle de donnes et ne doivent pas tre confondues avec les *versions* du modle de donnes, qui sont des ditions du modle voluant en parallle. Voir Gestion des versions du modle de donnes embarqu [p 93] pour plus d'informations sur les versions des modles de donnes.

Consultation et cration des publications

Pour accder aux publications existantes du modle de donnes courant, slectionnez "Grer les publications" dans son menu **"Actions" du modle de donnes [p 35]** dans le panneau de navigation. Vous pourrez consulter les dtails des publications et en crer de nouvelles.

Dans certains cas, il faudra utiliser plusieurs publications du mme modle de donnes afin de permettre diffrents jeux de donnes d'tre bass sur des tats diffrents du modle. L'utilisation de plusieurs publications doit tre ralise avec prcaution. En effet, les utilisateurs devront slectionner la publication mettre jour lors de la publication du modle de donnes et doivent donc avoir connaissance de l'utilisation de ces diffrentes publications. La cration d'une nouvelle publication est disponible uniquement pour les utilisateurs ayant le rle "Administrateur".

Pour crer une nouvelle publication, slectionnez "Grer les publications" dans le menu **"Actions" du modle de donnes [p 35]** dans le panneau de navigation, puis cliquez sur le bouton **Crer une publication**. Le nom donn la publication doit tre unique dans le rfrentiel. Aprs sa cration, la nouvelle publication est vide et ne reprsente pas un modle de donnes. Pour pouvoir tre utilise par des jeux de donnes, la publication doit tre mise jour en publiant le modle de donnes.

CHAPITRE 13

Gestion des versions de modles de donnes

Ce chapitre contient les sections suivantes :

1. A propos des versions
2. Accs aux versions
3. Actions sur les versions
4. Limitations connues sur la gestion de versions du modle de donnes

13.1 A propos des versions

Il est possible de crer des *versions* pour les modles de donnes, leur permettant ainsi d'voluer en parallle. Les versions ne doivent pas tre confondues avec les images des modles de donnes, qui sont prises au moment de publication et sont sauvegardes uniquement pour consultation d'historique en lecture seule.

13.2 Accs aux versions

Pour voir les versions existantes de votre modle de donnes, slectionnez "Grer les versions" dans le menu **"Actions" du modle de donnes** [p 35].

Les versions existantes sont reprsentes dans une arborescence selon leurs relations parent-enfant. Chaque modle de donnes a une version racine par dfaut.

13.3 Actions sur les versions

Dans l'espace de travail, en utilisant le menu avec la flche vers le bas ct de chaque version, accdez aux actions suivantes :

Accder la version du modle	Pour visualiser la version correspondante du modle de donnes.
Crer une nouvelle version	Cre une nouvelle version base sur le contenu de la version slectionne. La nouvelle version est cre comme enfant de la version slectionne, mais les contenus voluent indpendamment.
Dfinir en tant que version par dfaut	Dfinit la version par dfaut slectionne lorsqu'un utilisateur accde au modle de donnes.
Exporter une archive	Exporte le modle de donnes slectionn dans une archive contenant les donnes de la version, ses permissions et ses informations. L'archive est exporte vers le rpertoire d'archives, accessible par les administrateurs du rfrentiel. Voir Dossier archives [p 400] pour plus d'informations sur le rpertoire d'archives.
Importer une archive	Importe le contenu d'une archive dans la version slectionne. L'archive importer doit contenir un modle de donnes avec le mme nom que le modle associ cette version.

Une version peut tre supprime en cliquant le bouton **X** situ sa droite. Une version ne peut pas tre supprime si elle est lie une publication ou si elle a des sous versions. La version racine du modle de donnes ne peut pas tre supprime.

Deux versions du mme modle peuvent tre compares dans l'espace de travail en slectionnant leur case cocher, puis **"Actions" du modle de donnes > Comparer les versions slectionnes**. La vue de comparaison cte--cte affiche les diffrences structurelles entre les versions du modle de donnes, avec la plus ancienne gauche et la plus rcente droite.

13.4 Limitations connues sur la gestion de versions du modle de donnes

- Il est impossible de fusionner deux versions d'un modle de donnes.
- L'interface de comparaison n'affiche pas les mises jour des champs, uniquement les ajouts et suppressions.
- Il n'est pas possible de grer des versions de modles de donnes publis dans un module.
- Les ressources contenues dans un module et utilises par un modle de donnes embarqu ne sont pas versionnes lorsqu'une version est cre. En effet, seules les rfrences des ressources sont sauvegardes,

et les dveloppeurs doivent s'assurer que les ressources rfrences restent compatibles avec les diffrentes versions.

Espaces de données

CHAPITRE **14**

Introduction aux espaces de donnes

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Utilisation de l'interface utilisateur de la section Espace de donnes

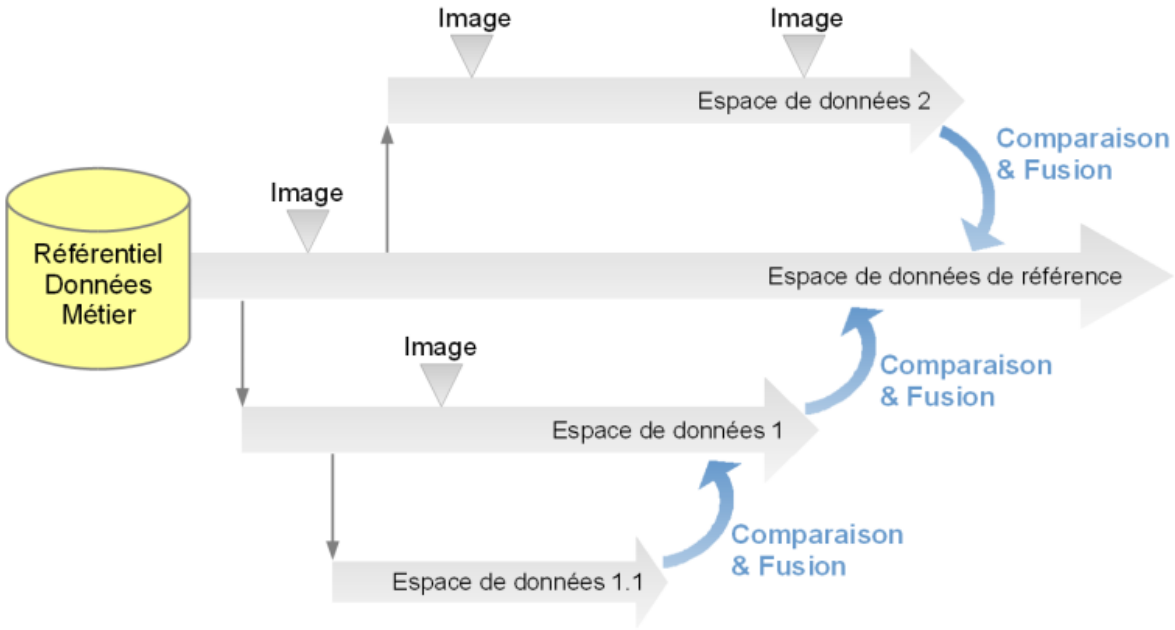
14.1 Prsentation

Fonction d'un espace de donnes

Le cycle de vie des donnes est souvent complexe. Il est parfois ncessaire d'entretenir une version courante des donnes tout en travaillant sur des volutions futures. De plus, il faut conserver une trace des tats intermdiaires. TIBCO EBX rend ces dmarches possibles grce aux espaces de donnes et aux images.

Un espace de donnes est un conteneur qui isole diffrentes versions de jeux de donnes et les organise. Des espaces de donnes enfants peuvent tre crs partir d'un espace de donnes. Un espace de donnes enfant est initialis avec le mme tat que son parent au moment de sa cration. Ainsi, des modifications peuvent tre effectues isolment dans l'espace de donnes enfant, sans impacter l'espace de donnes parent ni d'autres espaces de donnes. Lorsque les modifications dans l'espace de donnes enfant sont termines, cet espace de donnes peut tre fusionn avec son parent.

Une image est une copie statique d'un espace de donnes qui capture son tat et tout son contenu un moment donn. Les images peuvent tre consultes, exportes, et compares d'autres espaces de donnes.



Concepts de base lis aux espaces de donnes

La comprhension des termes suivants est recommande pour l'utilisation des espaces de donnes :

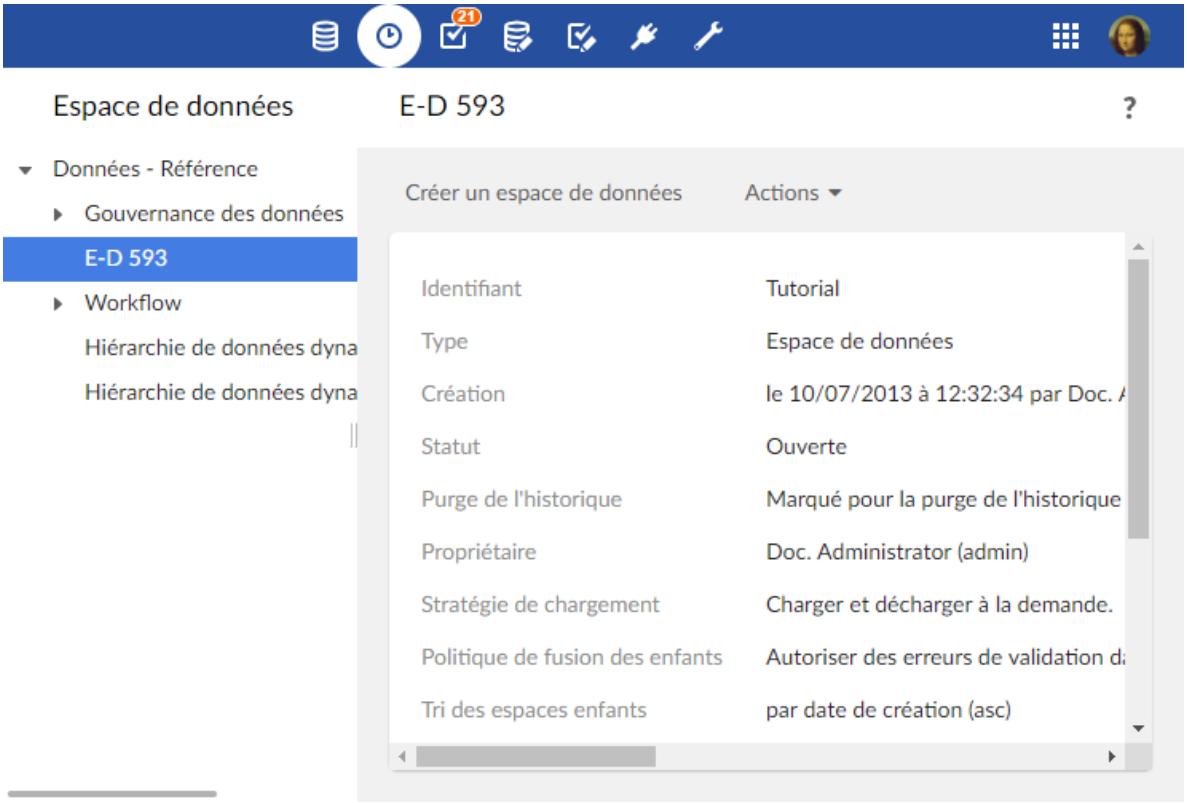
- espace de donnes [p 28]
- image [p 28]
- jeu de donnes [p 26]
- fusion [p 28]
- espace de donnes de rfrence [p 28]

14.2 Utilisation de l'interface utilisateur de la section Espace de donnes

Les espaces de donnes sont crs, consults et modifis dans la section **Espaces de donnes**.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance'.

Le panneau de navigation affiche l'organisation hirarchique des espaces de donnes existants, tandis que l'espace de travail affiche les informations concernant l'espace de donnes slectionn et liste ses images.



Voir aussi

Cration d'un espace de donnes *[p 101]*

Images *[p 113]*

Concepts apparentés*Jeux de donnes* *[p 120]*

CHAPITRE **15**

Cration d'un espace de donnes

Ce chapitre contient les sections suivantes :

1. Cration d'un espace de donnes
2. Proprits
3. Mode relationnel

15.1 Cration d'un espace de donnes

Par dfaut, les espaces de données dans TIBCO EBX sont en *mode smantique*. Ce mode supporte toutes les fonctionnalits permettant la gestion du cycle de vie des données.

Pour crer un nouvel espace de données en mode smantique, slectionnez un espace de données existant sur lequel il se basera, puis cliquez sur le bouton **Crer un espace de données** situ dans l'espace de travail.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance'.

L'espace de données nouvellement cr devient un espace de données enfant de celui qui tait slectionn. Son contenu est automatiquement initialis avec le contenu de son parent au moment de la cration. Une image initiale reprsentant cet tat est cre.

Except l'espace de données de rfrence, qui est la racine de tous les espaces de données smantiques du rfrentiel, les espaces de données smantiques sont toujours l'enfant d'un autre espace de données.

Voir aussi*Mode relationnel* *[p 102]*

15.2 **Proprits**

Les informations suivantes sont requises lors de la cration d'un nouvel espace de donnes:

Identifiant	Identifiant unique de l'espace de donnes.
Propritaire	Utilisateur possdant l'espace de donnes et qui est autoris en modifier les informations et les permissions. Le propritaire d'un espace de donnes n'est pas obligatoirement son crateur.
Libell	Libell et description associs l'espace de donnes en plusieurs langues.
Mode relationnel	Spcifie si l'espace de donnes est en mode relationnel. Cette option existe seulement lors de la cration d'un espace de donnes partir de l'espace de donnes de rfrence.

15.3 **Mode relationnel**

Les espaces de donnes en mode relationnel ne peuvent tre crs qu' partir de l'espace de donnes de rfrence. Ils sont limits en fonctionnalits par rapport aux espaces de donnes en mode smantique. Par exemple, les espaces de donnes en mode relationnel n'ont pas d'images et ne peuvent pas avoir d'espaces de donnes enfants.

Le mode relationnel implique que les tables de donnes de cet espace soient stockes directement dans un SGBDR.

Voir aussi*Mode relationnel* *[p 267]*

CHAPITRE 16

Actions sur les espaces de donnes existants

Ce chapitre contient les sections suivantes :

1. Informations de l'espace de donnes
2. Permissions sur un espace de donnes
3. Fusion d'un espace de donnes
4. Comparaison d'un espace de donnes
5. Validation d'un espace de donnes
6. Archives d'espaces de donnes
7. Fermeture d'un espace de donnes

16.1 Informations de l'espace de donnes

Certaines proprits associes un espace de donnes peuvent tre modifies en slectionnant **Actions** > **Information** dans l'espace de travail de la section Espaces de Donnes.

Documentation	Libell et description localiss associs l'espace de donnes.
Stratgie de chargement	*Seul l'administrateur a le droit de modifier cette option.* • **Chargement et dchargement sur demande :** le principal avantage de cette stratgie est de permettre de librer de la mmoire si ncessaire. L'inconvnient est que cela implique un cot de chargement quand on accde une ressource pour la premire fois depuis le dmarrage du serveur, ou si elle a t dcharge entre-temps. Ce mode est utilis par dfaut. • **Chargement forc :** ce mode est particulirement recommand pour les espaces de donnes et images utiliss de faon intensive, ou exigeants en termes de temps de rponse. • **Chargement forc et pr-validation :** ce mode est particulirement recommand pour les espaces de donnes et images utiliss de faon intensive ou exigeants en termes de temps de rponse, et pour lesquels le processus de validation peut tre long. **Note :** Toute modification de la stratgie de chargement requiert un redmarrage du serveur.
Politique de fusion	La politique de fusion des espaces de donnes enfants ne s'applique qu'aux fusions inities par un utilisateur. Elle ne s'applique pas aux fusions programmatiques, telles que celles inities par les scripts de workflow. Les politiques possibles sont : • **Autoriser des erreurs de validation dans le rsultat :** c'est la politique par dfaut. Un espace de donnes enfant peut tre fusionn quelle que soit la validit du rsultat de cette fusion. • **Fusion pr-validante :** un espace de donnes enfant ne peut tre fusionn que si le rsultat de la fusion est valide.
Propritaire	L'utilisateur possdant l'espace de donnes et qui est autoris en modifier les informations et les permissions. Le propritaire d'un espace de donnes n'est pas obligatoirement son crateur.

Tri des espaces enfants	Dfinit l'ordre d'affichage des espaces de donnes enfants dans l'arbre des espaces de donnes. Si la valeur n'est pas dfinie, l'ordre dfini par le parent est pris en compte. La valeur par dfaut est "par libell".
Changement de propritaire	Dfinit si le propritaire de l'espace de donnes a le droit de modifier l'attribut "Propritaire". Si la valeur est "Non habilit", seul l'administrateur a le droit d'effectuer cette modification.
Changement des permissions	Dfinit si le propritaire de l'espace de donnes a le droit de modifier les permissions de l'espace de donnes. Si la valeur est "Non habilit", seul l'administrateur a le droit d'effectuer cette modification.

16.2 Permissions sur un espace de donnes

Permissions gnrales

Identifiant de l'espace de donnes	Indique l'espace de donnes auquel les permissions vont tre appliques.
Slection du profil	Indique le profil concern par la permission dfinie.
Restriction d'accs	Indique si la permission dfinie restreint les permissions affectes un utilisateur donn par des politiques dfinies pour d'autres profils. Voir Restriction d'accs [p 307].
Permissions d'accs l'espace de donnes	Indique la permission globale d'accs l'espace de donnes. **Lecture seule** • Permet de visualiser l'espace de donnes et son image, et de voir les espaces de donnes enfants selon les droits s'appliquant chacun d'eux. • Permet de visualiser le contenu de l'espace de donnes sans pouvoir le modifier, condition que les droits s'appliquant au contenu en autorisent l'accs. **Ecriture** • Permet de visualiser l'espace de donnes et son image, et de voir les espaces de donnes enfants selon les droits s'appliquant chacun d'eux. • Permet de modifier le contenu de l'espace de donnes, condition que les droits s'appliquant au contenu en autorisent l'accs. **Non visible** • Ni l'espace de donnes, ni ses images ne peuvent tre vus. • A partir d'un espace de donnes enfant, l'espace de donnes courant est visible sans toutefois pouvoir tre slectionn. • Le contenu de l'espace de donnes n'est pas accessible. • Aucune action ne peut tre effectue sur l'espace de donnes.

Actions possibles

Les utilisateurs peuvent tre autoriss effectuer les actions suivantes :

Crer un espace de donnes enfant	Indique si le profil peut crer un espace de donnes enfant.
Crer une image	Indique si le profil peut crer une image de l'espace de donnes.
Initier une fusion	Indique si le profil peut fusionner un espace de donnes avec son parent.
Exporter une archive	Indique si le profil peut exporter l'espace de donnes.
Importer une archive	Indique si le profil peut importer dans l'espace de donnes.
Fermer un espace de donnes	Indique si le profil peut fermer l'espace de donnes.
Fermer une image	Indique si le profil peut fermer les images de l'espace de donnes.
Droits sur les services	Spcifie les permissions d'accs aux services.
Permissions des espaces de donnes enfants la cration	Spcifie les permissions d'accs aux espaces de donnes enfants crs partir de l'espace de donnes courant.

16.3 Fusion d'un espace de donnes

Quand le travail dans un espace de donnes est termin, il est possible d'effectuer une fusion unidirectionnelle de l'espace de donnes vers son espace de donnes parent. Le processus de fusion est le suivant :

1. l'espace de donnes parent et l'espace de donnes enfant sont verrouills pour tous les utilisateurs, except l'utilisateur qui a initi la fusion ainsi que les utilisateurs administrateurs. Les verrous sont maintenus pendant la dure de la fusion. Ainsi, les contenus des deux espaces de donnes peuvent tre lus, mais ne peuvent pas tre modifis.

   Note : en plus de s'appliquer aux modifications directes sur l'espace de donnes, cette restriction s'applique galement aux autres fusions des autres espaces de donnes enfants. Ainsi, il est impossible de fusionner d'autres espaces de donnes enfants tant que la fusion en cours n'est pas termine.
2. les changements qui ont t effectus dans l'espace de donnes depuis sa cration sont intgrs dans l'espace de donnes parent ;
3. l'espace de donnes enfant est ferm ;
4. l'espace de donnes parent est dverrouill.

Initiation d'une fusion

Suivez ces tapes pour initier la fusion d'un espace de donnes avec son espace de donnes parent :

1. slectionnez l'espace de donnes fusionner dans le panneau de navigation de la section Espaces de donnes ;
2. dans l'espace de travail, slectionnez **Fusionner l'espace de donnes** dans le menu **Actions**.

Revue et acceptation des changements

Avant la fusion dfinitive, slectionnez les changements survenus dans l'espace de donnes enfant (source) qui doivent tre fusionns dans l'espace de donnes parent (cible).

> **Note**
>
> Cette tape de revue n'existe pas pour une fusion faite travers un service de donnes ou un service programmatique:pour les fusions automatises, tous les changements dans l'espace de donnes enfant remplacent les donnes dans l'espace de donnes parent.

Pendant la fusion, un cran de comparaison rcapitule tous les changements devant tre revus. Deux colonnes de *listes de changements* sont affiches, prenant en compte les changements des comparaisons d'espaces de donnes suivantes :

- l'espace de donnes enfant par rapport son image initiale ;
- l'espace de donnes parent par rapport l'image initiale de l'espace de donnes enfant.

Par dfaut, tous les changements dtects sont slectionns pour la fusion. Pour exclure un changement de la fusion, dslectionnez-le. Les changements relatifs aux diffrents lments sont visibles en les slectionnant dans le panneau de navigation.

Afin de dtecter les conflits, la fusion implique l'espace de donnes courant, l'image initiale et l'espace de donnes parent. En effet, les donnes peuvent avoir t modifies la fois dans l'espace de donnes courant et dans son parent.

Le processus de fusion concerne aussi les droits d'accs aux tables. Il faut galement passer en revue les modifications des permissions pour dcider si elles seront incluses dans la fusion.

Lorsque vous avez choisi les changements fusionner, vous devez cliquer sur le bouton **Marquer les diffrences comme revues** pour indiquer que vous avez verifi les changements dans le primtre actuel. Tous les changements doivent tre revus pour effectuer la fusion.

Types de modifications

Le processus de fusion considre les oprations suivantes comme des modifications valuer :

- la cration d'un enregistrement ou d'un jeu de donnes ;
- la modification de toute entit ;
- la suppression d'un enregistrement, d'un jeu de donnes, ou de la valeur d'un noeud ;
- la modification des permissions d'une table.

Types de conflits

Cette interface de revue affiche galement les conflits dtects. Des conflits peuvent survenir quand une entit contient des modifications dans l'espace de donnes source et l'espace de donnes cible.

Les conflits sont catgoriss comme suit :

- conflit de cration d'un enregistrement ou d'un jeu de donnes,
- conflit de modification de toute entit,
- conflit de suppression d'un enregistrement ou d'un jeu de donnes,
- tout autre conflit.

Finalisation d'une fusion

Lorsque tous les changements ont t vrifis et que ceux inclure dans le rsultat de la fusion ont t choisis, cliquez sur le bouton **Fusionner >>** dans le panneau de navigation.

En fonction de la politique de fusion de l'espace de donnes parent, le processus de finalisation peut diffrer. Par dfaut, une fusion peut tre effectue mme si le rsultat de la fusion contient des erreurs de validation. En revanche, l'administrateur de l'espace de donnes parent peut configurer la politique de fusion pour que les fusions de ses espaces de donnes enfants soient finalises uniquement si le rsultat ne contient pas d'erreur de validation.

Si la politique de fusion pr-validante est utilise, un espace de donnes ddi est d'abord cr pour accueillir le contenu de la fusion. S'il est valide, cet espace de donnes est automatiquement fusionn avec l'espace de donnes parent.

Dans le cas contraire, si des erreurs de validation sont dtectes dans l'espace de donnes ddi, seuls seront accessibles l'espace de donnes d'origine et l'espace de donnes ddi contenant le rsultat de la fusion, nomm "[fusion] <*nom de l'espace de donnes enfant*>". Les options suivantes sont alors proposes dans le menu de l'espace de travail **Actions > Fusion en cours**:

- **Abandonner la fusion**:cette action abandonne la fusion en cours et restitue l'espace de donnes enfant dans son tat prcdant la fusion.
- **Poursuivre la fusion**:cette action permet de tenter de nouveau la fusion aprs avoir effectu les corrections ncessaires dans l'espace de donnes de fusion ddi.

Configuration de la politique de fusion d'un espace de donnes

En tant qu'administrateur d'un espace de donnes, il est possible, travers l'interface utilisateur, d'interrompre la finalisation des fusions de ses espaces de donnes enfants si le rsultat contient des erreurs de validation. Pour ce faire, cliquez sur **Actions > Informations** dans l'espace de travail de l'espace de donnes parent. Sur la page des informations de cet espace de donnes, positionnez la **Politique de fusion des enfants** "Fusion pr-validante". Cette politique de fusion sera applique pour toutes les fusions des espaces de donnes enfants avec l'espace de donnes parent ainsi paramtr.

> **Note**
>
> Si la fusion est effectue travers un composant web, le comportement pour la politique de fusion est le mme;la politique dfinie par l'espace de donnes parent sera automatiquement utilise lors de la fusion des modifications de l'espace de donnes enfant. Cependant, cette politique n'est pas applique pour les fusions programmatiques. Elle est donc ignore pour les tches automatiques des workflows de donnes.

Voir aussi*Politique de fusion* *[p 109]*

Abandon d'une fusion

Une fusion est effectue dans le contexte d'une session utilisateur et doit tre acheve en une seule opration. Si vous dcidez de ne pas poursuivre la fusion aprs l'avoir initie, cliquez sur le bouton **Annuler** afin d'abandonner l'opration.

Si vous naviguez vers une autre page pendant une fusion, celle-ci sera abandonne mais les verrous sur l'espace de donnes parent et l'espace de donnes enfant seront maintenus. Il faudra les dverrouiller dans la section **Espaces de Donnes**.

Pour dverrouiller un espace de donnes, slectionnez l'espace de donnes dans le panneau de navigation, et cliquez sur le bouton **Dverrouiller** dans l'espace de travail. Si vous effectuez le dverrouillage depuis l'espace de donnes enfant, les deux espaces de donnes seront dverrouills. Si vous effectuez le dverrouillage depuis l'espace de donnes parent, lui seul sera dverrouill, vous devrez donc aussi effectuer un dverrouillage de l'espace de donnes enfant.

16.4 Comparaison d'un espace de donnes

Il est possible de comparer le contenu d'un espace de donnes celui d'un autre espace de donnes ou une image dans le rfrentiel. Pour effectuer une comparaison, ouvrez l'espace de donnes dans le panneau de navigation, puis slectionnez **Actions > Comparer** dans l'espace de travail. L'assistant permet de choisir un autre espace de donnes ou une image comparer avec l'espace de donnes courant.

Pour une comparaison plus rapide, qui ignore les champs avec une valeur hrite ou calcule, slectionnez le filtre "Valeurs persistantes seulement".

Voir aussi*Comparer deux contenus* *[p 232]*

16.5 Validation d'un espace de donnes

Le contenu d'un espace de donnes peut tre valid globalement en utilisant le service de validation au niveau de l'espace de donnes. Ce service est accessible en slectionnant **Actions > Validation** dans l'espace de travail.

> **Note**
>
> Ce service est propos uniquement si l'utilisateur a la permission de valider tous les jeux de donnes contenus dans l'espace de donnes.

16.6 Archives d'espaces de donnes

Export d'une archive

Le contenu d'un espace de donnes peut tre export dans une archive en slectionnant **Actions > Export** dans le panneau de navigation. L'archive est sauvegarde sur le systme de fichiers du serveur, o seul un administrateur peut la rcuprer.

> **Note**
>
> Voir Archives directory [p 400] dans le Guide d'administration.

Pour raliser un export, les informations suivantes sont requises:

Nom du fichier archive	Nom de l'archive exporte.
Type d'export	Obligatoire. Le type d'export par dfaut est "Contenu complet de l'espace de donnes". Il permet d'exporter l'ensemble des donnes slectionnes dans l'archive. Il peut tre utile d'inclure uniquement les diffrences entre l'espace de donnes et son image initiale dans l'archive. Il existe deux types d'export qui incluent un delta:"Mises jour avec leur contenu complet" et "Mises jour seules". Le premier exporte l'tat actuel de l'espace de donnes ainsi que le delta qui contient les diffrences entre l'tat actuel de l'espace de donnes et l'image initiale. Le second exporte uniquement le delta qui contient les diffrences entre l'tat actuel de l'espace de donnes et l'image initiale. Les deux options affichent une page de comparaison, sur laquelle il est possible de slectionner les diffrences inclure dans le delta. Les diffrences sont dtectes au niveau table.
Jeu de donnes exporter	Jeux de donnes de cet espace de donnes exporter. Pour chaque jeu de donnes, il est possible de spcifier si les donnes, les permissions et les informations doivent tre exportes.

Importer une archive

Le contenu d'une archive peut tre import dans un espace de donnes en slectionnant **Actions > Import**.

Si l'archive slectionne ne contient pas de delta, l'tat actuel de l'espace de donnes sera remplac par le contenu de l'archive.

Si l'archive slectionne se compose d'un contenu complet et d'un delta, vous pouvez choisir d'appliquer le delta afin de fusionner les diffrences incluses. Un cran de comparaison sera affich, depuis lequel les diffrences fusionner peuvent tre slectionnes.

Si l'archive slectionne contient uniquement un delta, il est possible de slectionner les diffrences fusionner dans un cran de comparaison.

16.7 **Fermeture d'un espace de donnes**

Si un espace de donnes n'est plus utilis, il peut tre ferm. Ds qu'il est ferm, l'espace de donnes n’apparaîtra plus dans la section **Espaces de Donnes** de l'interface utilisateur, et ne sera plus accessible.

Un administrateur peut rouvrir un espace de donnes ferm, tant que celui-ci n'a pas encore t purg du rfrentiel.

Pour fermer un espace de donnes, slectionnez **Actions > Fermer l'espace de donnes**.

Voir aussi*Closing unused dataspaces and snapshots* *[p 402]*

CHAPITRE 17

Images

Ce chapitre contient les sections suivantes :

1. Prsentation des images
2. Cration d'une image
3. Visualisation de contenu d'une image
4. Informations de l'image
5. Comparaison d'une image
6. Validation d'une image
7. Export d'une image
8. Fermeture d'une image

17.1 Prsentation des images

Une image est une copie en lecture seule d'un espace de donnes. Une image sert de rfrence de l'tat et du contenu d'un espace de donnes un instant donn.

Voir aussi*image [p 28]*

17.2 Cration d'une image

Une image est cre en slectionnant un espace de donnes dans le panneau de navigation, puis en cliquant sur le bouton 'Crer une image' sous le menu 'Actions'.

Les informations suivantes sont requises :

Identifiant	Identifiant unique de l'image. Le modle suivant doit tre respect: [a-zA-Z0-9_:.\-|]*.
Libell	Libell et description localiss associs l'image.

17.3 **Visualisation de contenu d'une image**

Pour visualiser le contenu d'une image, ouvrir l'image, puis slectionner *Actions > Voir ou diter les jeux de donnes* dans l'espace de travail.

17.4 **Informations de l'image**

Certaines proprits associes une image peuvent tre modifies en slectionnant l'image, puis *Actions* > *Informations* dans l'espace de travail de la section *Espaces de donnes*.

Documentation	Libell et description localiss associs l'image.
Mode relationnel	Indique que les tables des jeux de donnes prsents dans cet espace de donnes sont contenues dans une base de donnes relationnelle. Il ne sera pas possible de crer des images et des espaces de donnes enfants.
Stratgie de chargement	Toute modification de ce champ requiert de redmarrer le serveur. Pour un espace de donnes en mode smantique, la stratgie par dfaut, 'Charger et dcharger la demande', permet de librer la mmoire si ncessaire. En contrepartie, cela implique un cot de chargement quand la ressource accde l'est pour la premire fois depuis le redmarrage du serveur ou quand elle a t dcharge depuis. La stratgie 'Chargement forc' est recommande pour les espaces de donnes et images vie longue et qui sont intensivement utilises. Pour un espace de donnes en mode relationnel, les stratgies sont 'Aucune' (dfaut) ou 'Prvalidation'. Seul l'administrateur a le droit de modifier cette option.Voir Stratgie de chargement [p 104].
Politique de fusion des enfants	Un espace de donnes peut dfinir une politique qui dfinit la manire de fusionner un espace de donnes enfant. La politique par dfaut est 'Autoriser des erreurs de validation dans le rsultat'. Une autre politique, la 'Fusion pr-validante', permet de s'assurer que le rsultat de la fusion est valide avant d'effectuer dfinitivement la fusion. La politique de fusion des enfants est uniquement applique en utilisant les interfaces utilisateur. Ce paramtrage est ignor pour une fusion programmatique (incluant les tches automatiques des workflows de donnes).
Propritaire	L'utilisateur possdant l'image, et qui est autoris en modifier les informations et les permissions. Le propritaire d'une image n'est pas obligatoirement son crateur.
Tri des espaces enfants	Dfinit l'ordre d'affichage des espaces de donnes enfants dans l'arbre des espaces de donnes. Si non dfini, l'ordre dfini par le parent est pris en compte. La valeur par dfaut est 'par libell'.

Changer le propritaire	Dfinit si le propritaire de l'image a le droit de modifier l'attribut "Propritaire". Si la valeur est "Non habilit", seul l'administrateur a le droit d'effectuer cette modification.
Changer les permissions	Spcifie si un utilisateur qui est propritaire de l'espace de donnes a le droit de modifier les permissions de cet espace de donnes. Si ce n'est pas le cas, seul un administrateur ou un 'super propritaire' de l'espace de donnes a le droit de modifier les permissions.

17.5 Comparaison d'une image

Il est possible de comparer le contenu d'une image celui d'une autre image ou d'un espace de donnes dans le rfrentiel. Pour effectuer une comparaison, ouvrir l'image, puis slectionner *Actions* > *Comparer* dans l'espace de travail. L'assistant permet de choisir une autre image ou un espace de donnes comparer avec l'image courante.

Pour une comparaison plus rapide, qui ignore les champs avec une valeur hrite ou calcule, slectionner le filtre 'Valeurs persistantes seulement'.

Voir aussi*Comparer deux contenus* *[p 232]*

17.6 Validation d'une image

Le contenu d'une image peut tre valid globalement en utilisant le service de validation au niveau de l'image. Ce service est accessible en slectionnant *Actions* > *Valider* dans l'espace de travail.

> **Note**
>
> Pour utiliser ce service, l'utilisateur doit avoir la permission de valider tous les jeux de donnes contenus dans l'image.

17.7 Export d'une image

Le contenu d'une image peut tre export dans une archive en slectionnant *Actions* > *Exporter* dans l'espace de travail. L'archive est sauvegarde sur le systme de fichiers du serveur, o seul un administrateur peut la rcuprer.

> **Note**
>
> Voir Archives directory [p 400] dans le Guide d'administration.

Pour raliser un export, les informations suivantes sont requises :

Nom du fichier archive crer	Le nom de l'archive exporte.
Type d'export	Le contenu complet de l'espace de donnes, Mises jour avec leur contenu complet (*), Mises jour seules (*). *(*):les mises jour exporter sont slectionnes dans les pages suivantes.*
Jeu de donnes (ou arbre de jeux de donnes) exporter	Les jeux de donnes de cette image exporter. Pour chaque jeu de donnes, il est possible de choisir si les donnes, les permissions et les informations doivent tre exportes.

17.8 Fermeture d'une image

Si une image n'est plus utilise, elle peut tre ferme. Ds qu'elle est ferme, l'image n’apparaît plus dans la section 'Espaces de donnes' de l'interface utilisateur et n'est plus accessible.

Un administrateur peut rouvrir une image ferme, tant qu'elle n'a pas t purge du rfrentiel.

Pour fermer une image, slectionner *Actions > Fermer l'image*.

Voir aussi*Closing unused dataspaces and snapshots* *[p 402]*

Jeux de données

CHAPITRE **18**

Introduction aux jeux de donnes

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Utilisation de l'interface utilisateur de la section Donnes

18.1 Prsentation

Fonction d'un jeu de donnes

Un jeu de donnes est un conteneur de donnes qui se base sur les dfinitions de structure fournies par le modle de donnes qu'il implmente. Lors de la publication d'un modle de donnes, il est possible de crer des jeux de donnes bass sur sa dfinition. Par la suite, si ce modle est modifi et republi, tous ses jeux de donnes associs sont mis jour automatiquement.

Dans un jeu de donnes, les valeurs de donnes sont consultables et modifiables. A l'aide des vues, il est possible d'afficher les tables d'une manire adapte la nature des donnes et du mode d'accs. Les recherches et les filtres peuvent aussi tre utiliss pour restreindre l'affichage ou rechercher des donnes.

Des permissions peuvent aussi tre affectes diffrents rles pour contrler l'accs au niveau du jeu de donnes. Ainsi, en appliquant des permissions spcifiques, on peut permettre certains utilisateurs d'afficher ou de modifier des donnes tout en les cachant d'autres.

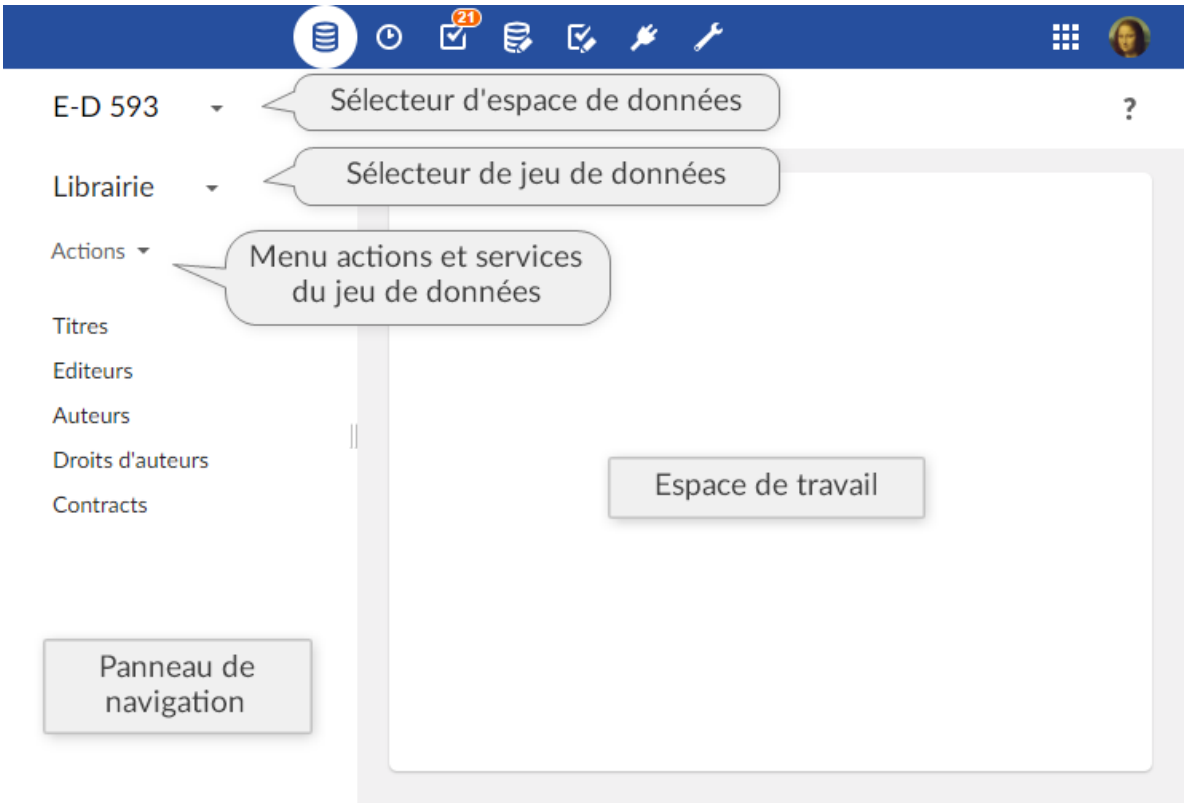
Concepts de base lis aux jeux de donnes

La comprhension des termes suivants est recommande pour utiliser les jeux de donnes :

- espace de donnes [p 28]
- jeu de donnes [p 26]
- enregistrement [p 26]
- champ [p 25]
- cl primaire [p 25]
- cl trangre [p 25]
- table (jeu de donnes) [p 26]
- groupe [p 25]

18.2 Utilisation de l'interface utilisateur de la section Donnes

Dans le cadre de l'utilisation de la Perspective avance [p 17], ou d'une perspective spcifique, les jeux de donnes sont crs, consults et modifis dans la section 'Donnes'. Seuls les utilisateurs autoriss peuvent accder ces interfaces spcifiques.



Pour slectionner ou crer un jeu de donnes, cliquer sur 'Slectionner le jeu de donnes' dans le panneau de navigation. La structure de donnes du jeu de donnes s'affichera dans le panneau de navigation et les formulaires d'enregistrement ainsi que les vues de table s'afficheront dans l'espace de travail.

Lors de la visualisation d'une table du jeu de donnes dans l'espace de travail, le bouton permet d'afficher les recherches et filtres disponibles pour restreindre l'affichage des enregistrements.

Les actions applicables au jeu de donnes sont disponibles dans le menu **Actions** du panneau de navigation (les services sont accessibles en bas de la liste).

Voir aussi

Cration du jeu de donnes [p 123]

Recherche et filtrage des donnes [p 126]

Actions sur les enregistrements dans l'interface utilisateur [p 135]

Hritage [p 27]

Concepts apparentés

Modle de donnes [p 34]

Espace de donnes *[p 98]*

CHAPITRE **19**

Cration du jeu de donnes

Ce chapitre contient les sections suivantes :

1. Cration d'un jeu de donnes racine
2. Cration d'un jeu de donnes enfant utilisant l'hritage

19.1 Cration d'un jeu de donnes racine

Afin de crer un jeu de donnes racine, qui n'hrite pas d'un jeu de donnes parent, il faut slectionner le menu 'Slectionner le jeu de donnes [p 121]' dans le panneau de navigation, cliquer sur le bouton 'Crer un jeu de donnes' dans la fentre contextuelle, et suivre l'assistant.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance' ou via une perspective spcifique.

L'assistant vous permet de slectionner un des trois types de modles de donnes, sur lequel le nouveau jeu de donnes sera bas:packag, embarqu ou externe.

- Un *modle de donnes packag* est un modle de donnes dfini l'intrieur d'un module (application web).
- Un *modle de donnes embarqu* est un modle de donnes cr et publi en utilisant l'assistant de modles de donnes. Il est entirement gr l'intrieur du rfrentiel TIBCO EBX. Un modle de donnes doit avoir t pralablement publi pour que cette fonctionnalit soit disponible.
- Un *modle de donnes externe* est un modle de donnes rfrenc au moyen d'une URI.

Aprs avoir choisi le modle de donnes sur lequel le nouveau jeu de donnes sera bas, vous devez fournir un nom unique, sans espaces ni caractres spciaux. Facultativement, vous pouvez donner des libells localiss pour le jeu de donnes, qui seront affichs aux utilisateurs dans l'interface en fonction de leurs prfrences de langue.

> **Attention**
>
> Le contenu des tables n'est pas copi lors de la duplication d'un jeu de donnes.

19.2 Cration d'un jeu de donnes enfant utilisant l'hritage

Le mcanisme d'hritage permet d'utiliser les relations parent-enfant, grce auxquelles les jeux de donnes enfants hritent par dfaut des valeurs de leurs jeux de donnes anctres. Pour pouvoir crer des jeux de donnes enfants depuis un jeu de donnes parent, il faut activer l'hritage entre jeux de donnes dans le modle de donnes associ.

Pour crer un jeu de donnes enfant, cliquer sur Crer ou slectionner un jeu de donnes [p 121] dans le panneau de navigation, puis sur le bouton + correspondant au jeu de donnes parent.

Le jeu de donnes enfant sera bas sur le mme modle de donnes que celui de son parent, et donc les seules informations spcifier sont son nom unique, et ventuellement les libells localiss.

Voir aussi*Hritage entre jeux de donnes [p 143]*

CHAPITRE 20

Visualisation des donnes

TIBCO EBX offre un mcanisme de personnalisation des tables via la fonctionnalit des vues. Une vue permet de spcifier les colonnes afficher ainsi que l'ordre d'affichage. Les vues peuvent tre gres par profil grce au concept de vues recommandes [p 131].

Ce chapitre contient les sections suivantes :

1. Menu 'Vue'
2. Tri des donnes
3. Recherche et filtrage des donnes
4. Vues
5. Gestion des vues
6. dition tabulaire
7. Historique

20.1 Menu 'Vue'

Le menu droulant 'Vue' permet d'accder facilement aux diffrentes vues disponibles et aux fonctionnalits de gestion des vues.

Les vues sont gres dans le sous-menu ddi 'Gestion de vues' [p 132].

Les vues peuvent galement tre regroupes. L'administrateur doit avoir dfini des groupes au pralable dans la table 'Groupes de vues' de la section 'Configuration des vues'. L'utilisateur peut ainsi dfinir une vue comme appartenant un groupe dans le champ 'Groupe de la vue' lors de la cration ou de la modification d'une vue. Voir Description d'une vue [p 128] pour plus d'informations.

20.2 Tri des donnes

Les critres de tri contrlent l'ordre dans lequel les enregistrements sont prsents.

Par dfaut, les enregistrements sont ordonns par cl primaire ascendante.

Afin de dfinir des critres de tri spcifiques, cliquer sur le bouton 'Slection et tri' dans l'espace de travail.

Chaque critre de tri est dfini par un nom de colonne et un ordre de tri (ascendant ou descendant). Utiliser les flches 'Dplacer gauche/droite' pour ajouter ou enlever un critre dans la table 'Tri'. Une fois un critre slectionn, il est possible de choisir son ordre de tri en cliquant sur le bouton 'ASC' ou 'DESC' droite.

Pour changer la priorit d'un critre de tri, slectionnez-le dans la liste, puis cliquez sur les flches vers le haut ou vers le bas pour le dplacer.

Pour rtablir la vue sans critre de tri spcifique, slectionner *Vue* > *Rtablir*.

20.3 Recherche et filtrage des donnes

Des outils spcifiques permettent la recherche d'enregistrements dans une table. On peut y accder par l'icne , qui affiche les panneaux des filtres.

Chaque panneau de filtre (ou de recherche) propose une case cocher sur sa barre de titre:cocher cette case applique le filtre;la dcocher le dsapplique.

> **Note**
>
> Appliquer une vue rinitialisera et retirera tous les filtres actuellement appliqus.

Recherche

En mode simple, la 'Recherche' permet d'ajouter des critres contextualiss sur un ou plusieurs champs. Lors de l'ajout d'un critre, les oprateurs proposs seront en adquation avec le type du champ correspondant.

En activant le mode avanc, il est possible de crer des sous-blocs contenant des critres, afin de crer des oprations logiques plus complexes dans l'laboration du filtre.

> **Note**
>
> En mode avanc, les critres dont l'oprateur est "correspond " ou "correspond (respecter la casse)" suivent la syntaxe standard des expressions rgulires de Java.

Voir aussi*Regex pattern*

Recherche textuelle

La recherche textuelle est utilise pour une recherche de texte brut sur un ou plusieurs champs. Cette recherche ne prend pas en compte le type du champ.

- Si le texte entr contient un ou plusieurs mots sans caractre de remplacement (`*` ou `?`), les champs trouvs seront ceux contenant tous ces mots. Les mots entre guillemets ("aa bb" par exemple) sont considrs comme un seul mot.
- Les caractres usuels de remplacement sont proposs:l'toile `*` (tout texte) ou le point d'interrogation `?` (tout caractre). Pour des raisons de performance, leur utilisation est restreinte un seul par recherche.
- Les caractres de remplacement peuvent tre considrs comme de simple caractres en les chappant l'aide du caractre '\', par exemple, '*'.

Exemples :

- `aa bb`:le champ contient "aa" et "bb".
- `aa "bb cc"`:le champ contient "aa" et "bb cc".
- `aa*`:le champ commence par "aa".
- `*bb`:le champ se termine par "bb".

- `aa*bb`:le champ commence par "aa" et se termine par "bb".
- `aa?`:le champ commence par "aa" et a une taille de 3 caractres.
- `?bb`:le champ se termine par "bb" et a une taille de 3 caractres.
- `aa?bb`:le champ commence par "aa" et se termine par "bb" et a une taille de 5 caractres.
- `aa\*bb`:le champ contient "aa*bb" tel quel.

Sur les tables de grande taille, il est recommand de ne slectionner qu'un seul champ de la table;si le champ n'est pas du type "chaîne de caractres", il est conseill d'entrer un texte au bon format, par exemple :

- boolen:Oui, Non
- date:01/01/2000
- nombre:100000 ou 100 000
- champ numr:Rouge, Bleu...

L'option *Sensible la casse* oblige la recherche tenir compte de la casse, en distinguant les majuscules des minuscules.

Recherche sur validation

La recherche par validation permet de voir les enregistrements en fonction de leur statut dans la dernire validation effectue. Il est possible de voir les enregistrements avec les niveaux 'Erreur', 'Avertissements' et 'Informations'.

> **Note**
>
> Cette recherche s'applique seulement aux enregistrements de tables qui ont dj t valids;pour ce faire, slectionner *Actions* > *Valider* au niveau de la table dans l'espace de travail, ou au niveau du jeu de donnes dans le panneau de navigation.

Recherches spcifiques sur tables

Pour chaque table, le modle peut spcifier des filtres additionnels pour la recherche.

Voir aussi*Spcifier les filtres UI sur une table [p 208]*

20.4 Vues

Il est possible de personnaliser l'affichage des tables dans EBX en fonction de l'utilisateur cible. Il existe deux types de vues:tabulaire [p 129] et hirarchique [p 129].

Une vue est cre en slectionnant *Vue* > *Crer une nouvelle vue* dans l'espace de travail. Pour appliquer une vue, la slectionner dans *Vue* > *nom de la vue* .

Deux modes avancs de visualisation sont disponibles la cration d'une nouvelle vue :

- 'Vue tabulaire simple':une vue tabulaire qui permet de trier et filtrer les enregistrements affichs ;
- 'Vue hirarchique':une arborescence qui lie les donnes de diffrentes tables en utilisant leurs relations.

Description d'une vue

Lors de la cration ou de la modification d'une vue, la premire page permet de dfinir les informations gnrales concernant la vue.

Documentation	Libell et description localiss associs la vue.
Propritaire	Nom du propritaire de la vue, qui peut ce titre la grer et la modifier. (Seulement disponible pour les administrateurs et le propritaire du jeu de donnes)
Partager avec	Autres profils pouvant utiliser cette vue depuis le menu 'Vue'. **Note** Requiert une permission, voir Permissions des vues [p 421].
Mode de vue	Vue tabulaire simple ou vue hirarchique.
Groupe de la vue	Groupe d'appartenance de cette vue (le cas chant).

Vue tabulaire simple

Les vues tabulaires simples offrent la possibilit de dfinir des critres pour filtrer les enregistrements et de slectionner les colonnes afficher.

Colonnes affiches	Spcifie, l'aide de flches, les colonnes de la table afficher dans la vue.
Colonnes tries	Spcifie l'ordre d'affichage des colonnes et indique si les enregistrements de chaque colonne sont tris par ordre croissant ou dcroissant. Voir Tri des donnes [p 125].
Filtre	Dfinit les critres utiliss pour filtrer les enregistrements. Voir Editeur de critres [p 317].
Limite de pagination	Force une limite au nombre d'enregistrements visibles.
dition tabulaire	Si active, les utilisateurs de cette vue pourront basculer en dition tabulaire afin d'diter des enregistrements directement depuis la vue tabulaire.
Dsactiver la cration et la duplication	Si oui, les utilisateurs de cette vue ne pourront ni crer ni dupliquer d'enregistrement depuis l'dition tabulaire.

Vues hirarchiques

Une hirarchie est une arborescence permettant de prsenter les relations existant entre les tables. Elle peut tre structure sur plusieurs niveaux, appels niveaux de dimension. En outre, il est possible de dfinir des filtres sur des niveaux afin de filtrer les enregistrements.

Dimension d'une hirarchie

Une dimension dfinit une dpendance dans la hirarchie. Par exemple, une dimension pourrait tre prcise pour afficher des produits par catgorie. Plusieurs dimensions peuvent tre dfinies pour une vue hirarchique.

Options de configuration d'une vue hirarchique

Ce formulaire permet de configurer les options d'une vue hirarchique.

Afficher les enregistrements dans une nouvelle fentre	Si "oui", une nouvelle fentre sera ouverte pour afficher l'enregistrement. Sinon, il sera affich dans une nouvelle page de la fentre courante.
Hirarchie lague	Si "oui", les noeuds de la hirarchie qui n'ont pas d'enfants et qui n'appartiennent pas la table cible ne seront pas affichs.
Afficher les orphelins	Si "oui", les noeuds de la hirarchie qui n'ont pas de parent seront affichs.
Afficher le noeud racine	Si 'Non', le noeud racine de la hirarchie sera cach de la vue.
Libell du noeud racine	Libell localis du noeud racine de la hirarchie.
Barre d'outils au dessus de la hirarchie	Permet de positionner la barre d'outils en haut de la vue hirarchique.
Afficher les enfants des noeuds recherchs	Dans le cas rcursif, quand un filtre de recherche est appliqu, dfinit si doivent tre affichs les noeuds enfants des noeuds vrifiant le filtre.
Retirer les noeuds racines rcursifs sans enfants	Dans le cas rcursif, quand un filtre de recherche est appliqu ou lorsque le mode est 'lagu', permet de ne pas afficher les noeuds racines sans enfants.
Dtecter les cycles	Dans le cas rcursif, autorise la dtection et l'affichage des noeuds appartenant un cycle, en choisissant comme racine le plus ancien noeud du cycle. Limite : ne fonctionne pas en mode recherche ou lagu.

Libells

Pour chaque niveau de dimension faisant rfrence une autre table, il est possible de dfinir les libells localiss pour les noeuds correspondants dans la hirarchie. Utiliser l'assistant pour slectionner les champs utiliss dans les dfinitions des libells.

Filtre

L'diteur de critres permet de dfinir un filtre d'enregistrement pour la vue.

Voir aussi*Editeur de critres* *[p 317]*

Champ d'ordonnancement

Afin de pouvoir dplacer les noeuds dans la vue hirarchique, il faut dsigner un champ d'ordonnancement ligible. Celui-ci est dfini dans la table sur laquelle est applique la vue hirarchique. Un champ

d'ordonnancement doit avoir le type de donnes 'Entier' et doit avoir la vue par dfaut 'Cach' dans les proprits avances du modle de donnes.

Des actions de positionnement sur chaque noeud sont alors possibles, moins que le champ d'ordonnancement ne soit en lecture seule ou qu'un filtre ne soit dfini sur la hirarchie.

En l'absence d'un noeud d'ordonnancement, les noeuds enfants sont tris selon l'ordre alphabtique des libells des noeuds.

Attention

Le champ d'ordonnancement doit tre un champ technique ddi, et non un champ contenant des donnes mtier. En effet, ces donnes seront modifies automatiquement par les actions de rordonnancement.

Actions sur un noeud de hirarchie

Chaque noeud d'une vue hirarchique possde un menu correspondant qui offre des actions contextuelles.

Les noeuds terminaux peuvent tre dtachs de leur parent en slectionnant l'option 'Dtacher du parent'. L'enregistrement devient ainsi un noeud orphelin dans l'arborescence, rang sous un conteneur portant le nom 'non dfini'.

Les noeuds terminaux peuvent aussi changer de noeud parent, grce l'option 'Attacher un autre parent'. Si, selon le modle de donnes, un noeud peut avoir plusieurs parents, le noeud sera la fois sous le parent d'origine et rajout galement sous le nouveau parent. Sinon, le noeud terminal sera dplac sous le nouveau noeud parent.

Partage de vues

Les utilisateurs disposant de la permission 'Partager des vues' sur une vue ont la possibilit de dfinir quels utilisateurs peuvent slectionner cette vue depuis leur menu 'Vue'.

Pour cela, il suffit d'ajouter des profils dans le champ 'Partager avec' de l'cran de configuration de la vue.

Publication de vue

Les utilisateurs disposant de la permission 'Publier des vues' peuvent publier les vues qui apparaissent dans leur menu 'Vue'.

Une vue publie devient accessible tous les utilisateurs via les composants Web, les tches utilisateur du workflow, les services de donnes et les perspectives. Pour publier une vue, slectionner 'Publier' dans le menu *Vue > Grer les vues > nom de la vue > Publier*.

20.5 **Gestion des vues**

Grer les vues recommandes

Le propritaire d'un jeu de donnes peut dfinir des vues recommandes pour chaque profil cible.

Quand un utilisateur se connecte sans spcifier de vue, la vue recommande, si elle existe, s'applique. Sinon, la vue par dfaut s'affiche. L'action 'Grer les vues recommandes' permet de dfinir les rgles d'attribution des vues recommandes par utilisateur et par rle.

Les actions disponibles sur les vues recommandes sont les suivantes : changer l'ordre d'attribution des rgles, ajouter une rgle, diter une rgle, supprimer une rgle existante.

Pour un utilisateur donn, les vues recommandes sont values en fonction du profil de l'utilisateur:la rgle applique sera la premire de la liste qui correspond au profil de l'utilisateur.

> **Note**
>
> La fonctionnalit 'Grer les vues recommandes' est uniquement accessible au propritaire du jeu de donnes.

Grer les vues

Le sous-menu 'Grer les vues' permet d'accder aux actions suivantes:

Dfinir cette vue comme ma favorite	Uniquement disponible lorsque la vue en cours d'affichage n'est PAS la vue recommande. La vue favorite sera automatiquement applique lors de l'accs la table.
Dfinir la vue recommande comme ma favorite	Uniquement disponible lorsqu'une vue favorite est dfinie. En cliquant sur cette action, la vue courante ne sera plus la vue favorite de l'utilisateur. Une vue recommande, l'instar de la vue favorite, est applique automatiquement lors de l'accs la table. **Cette action n'est pas affiche si aucune vue favorite n'a t dfinie.**

20.6 **dition tabulaire**

La fonctionnalit d'dition tabulaire permet de modifier les donnes dans une vue table. Elle est accessible en cliquant sur le bouton .

L'accs l'dition tabulaire partir d'une vue table ncessite que la fonctionnalit ait t pralablement active dans la configuration de la vue.

Voir aussi*dition tabulaire [p 129]*

Copier/coller

Le copier/coller d'une cellule vers une autre cellule de la mme table s'effectue grce au menu *dition*. Il est aussi possible d'utiliser le raccourci clavier associ *Ctrl+C* et *Ctrl+V*.

Ce systme n'utilise pas le presse-papier natif du systme d'exploitation mais un mcanisme interne. Copier une cellule et la coller dans un fichier externe ne fonctionnera donc pas. Inversement, coller une valeur dans une cellule de la table ne fonctionnera pas non plus.

Tous les champs de type simple utilisant les composants intgrs sont supports, sauf :

- les cls trangres pointant vers des champs qui ne sont pas des chaînes de caractres ;
- les numerations qui ne sont pas des chaînes de caractres.

20.7 **Historique**

La fonctionnalit d'historique permet le suivi des modifications de donnes.

La visualisation de l'historique de donnes ncessite que la fonctionnalit ait t pralablement active au niveau des tables dans le modle de donnes. Voir Proprits avances des tables [p 65] pour plus d'informations.

Pour visualiser l'historique d'un jeu de donnes, slectionner *Actions* > *Historique* dans le panneau de navigation.

Pour visualiser l'historique d'une table ou d'une slection d'enregistrements, slectionner *Actions* > *Voir l'historique* dans l'espace de travail.

Il existe diffrents modes d'historique qui permettent de visualiser l'historique des donnes selon diffrentes perspectives:

Historique dans l'espace de donnes en cours	La vue d'historique de table affiche les oprations sur la branche en cours. C'est le mode par dfaut.
Historique dans l'espace de donnes courant et ses anctres	La vue d'historique de table affiche les oprations sur la branche en cours ainsi que tous ses anctres.
Historique dans l'espace de donnes courant et ses enfants fusionns	La vue d'historique de table affiche les oprations sur la branche en cours ainsi que toutes ses branches filles fusionnes.
Historique dans tous les espaces de donnes	La vue d'historique de table affiche les oprations sur toute la hirarchie de la branche en cours.

Dans la vue d'historique, utiliser le menu *View* pour passer un autre mode de l'historique.

Voir aussi*Historique [p 273]*

CHAPITRE 21

Edition des donnes

Ce chapitre contient les sections suivantes :

1. Actions sur les enregistrements dans l'interface utilisateur
2. Import et export de donnes
3. Restauration depuis l'historique

21.1 Actions sur les enregistrements dans l'interface utilisateur

L'dition des enregistrements s'effectue dans l'espace de travail de l'interface utilisateur.

> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance' ou via une perspective spcifique.

Cration d'un enregistrement

Dans la vue tabulaire, un nouvel enregistrement peut tre cr l'aide du bouton **+** situ en haut gauche de la table.

Dans une vue hirarchique, slectionnez "Crer un enregistrement" dans le menu du noeud parent du nouvel enregistrement.

Dans les deux cas, un formulaire s'affiche, permettant d'entrer des donnes. Les donnes obligatoires sont repres par une astrisque rouge.

Modification d'un enregistrement

Un enregistrement peut tre dit par double clic. Le formulaire qui s'affiche permet d'diter l'enregistrement, tandis que le bouton *Rtablir* permet de recharger le formulaire sans soumettre aucun des changements effectus.

Dupliquer un enregistrement

Pour dupliquer un enregistrement, slectionnez-le, puis slectionnez *Actions* > *Dupliquer*.

Un formulaire apparaît, pr-rempli partir des valeurs de l'enregistrement copi. La cl primaire doit ensuite tre modifie pour pouvoir crer ce nouvel enregistrement, moins qu'elle ne soit gnre automatiquement (l'exemple d'une valeur auto-incrmente).

Supprimer

Pour supprimer un ou plusieurs enregistrements slectionns, slectionnez *Actions* > *Supprimer*.

Comparer

Deux enregistrements slectionns peuvent tre compars en slectionnant *Actions* > *Comparer*.

> **Note**
>
> La comparaison ne prend pas en compte le contenu des noeuds terminaux complexes, comme les listes aggrges ou les attributs utilisateurs. Toutes les diffrences portant sur de tels noeuds seront ignores.

21.2 Import et export de donnes

Dans une table, les enregistrements peuvent tre imports ou exports, depuis ou vers les formats CSV ou XML.

Vous pouvez soit exporter l'ensemble de la table, soit slectionner manuellement certains enregistrements exporter, grce aux cases cocher.

Voir aussi

Services CSV *[p 241]*

Services XML *[p 235]*

21.3 Restauration depuis l'historique

Dans une table o l'historique est activ, il est possible de restaurer un enregistrement un tat antrieur en se basant sur son historique. Dans le cas o l'enregistrement (identifi par sa cl primaire) existe encore dans la table, il sera mis jour par les valeurs historises restaurer. Dans le cas contraire, il sera cr.

Pour restituer un enregistrement un tat antrieur, slectionner un enregistrement dans la vue d'historique, puis slectionner *Actions* > *Restaurer depuis l'historique* dans l'espace de travail. Un cran de rsum s'affiche avec les dtails de la mise jour ou de la cration effectuer.

La fonctionnalit de restauration est uniquement applicable sur un enregistrement la fois.

Si un trigger de table doit avoir un comportement spcifique pour la restauration, c'est dire diffrent de celui des oprations classiques de cration et de mise jour, le dveloppeur peut utiliser la mthode `TableTriggerExecutionContext.isHistoryRestore`[API].

> **Note**
>
> Il existe des limitations lies la fonctionnalit de l'historique :
>
> - La restauration depuis l'historique n'est pas disponible pour les tables contenant des listes non supportes par l'historique. Voir Limitations du modle de donnes [p 278].
> - Les types de champ suivants ne sont pas historiss:les valeurs calcules, les champs crypts, ainsi que les champs dont l'historique est dsactiv. Ces champs seront donc ignors lors de la restauration d'un enregistrement depuis l'historique.

Voir aussi*Historique* *[p 273]*

CHAPITRE 22

Actions sur les jeux de donnes existants

Ce chapitre contient les sections suivantes :

1. Validation du jeu de donnes
2. Duplication d'un jeu de donnes
3. Dsactivation d'un jeu de donnes
4. Gestion des permissions de jeux de donnes

22.1 Validation du jeu de donnes

Il est possible de valider un jeu de donnes en slectionnant *Actions* > *Valider* depuis le panneau de navigation. Les ventuels messages issus de la validation du jeu de donnes sont prsents dans un rapport. Depuis le rapport de validation, cliquer sur le bouton *Revalider* pour mettre jour ce rapport. Pour supprimer tous les messages de validation actuellement associs au jeu de donnes, et pouvoir relancer une validation complte, cliquer sur le bouton *Rinitialiser le rapport de validation*.

Dans la section 'Donnes', il est galement possible de valider une table, en la slectionnant dans le panneau de navigation et en utilisant l'action *Actions* > *Valider* dans l'espace de travail.

Voir Validation [p 324] pour obtenir plus d'informations concernant la validation incrmentale de donnes.

22.2 Duplication d'un jeu de donnes

Pour dupliquer un jeu de donnes existant, slectionnez-le dans le menu "*Slectionner le jeu de donnes* [p 121]" dans le panneau de navigation, puis slectionnez *Actions* > *Dupliquer*.

22.3 Dsactivation d'un jeu de donnes

Si un jeu de donnes est activ, il sera sujet la validation. Tous les lments obligatoires doivent tre dfinis pour que le jeu de donnes soit valide. Si un jeu de donnes est activ et valide, on considre qu'il peut tre export de faon scurise vers des systmes externes (ou utilis par d'autres applications Java).

Il est possible de dsactiver un jeu de donnes dont les lments obligatoires ne sont pas dfinis, en spcifiant "Non" pour la proprit "Activ" dans *Actions* > *Informations*.

22.4 Gestion des permissions de jeux de donnes

Les permissions de jeux de donnes sont accessibles en slectionnant *Actions* > *Permissions* dans le panneau de navigation.

Les permissions sont dfinies en crant des *profils*. Pour crer un nouveau profil de permissions, crez un nouvel enregistrement dans la table "Droits d'accs par profil".

Voir aussi*Profil* *[p 23]*

Profil	Indique le profil concern par la permission dfinie.
Restriction d'accs	Indique si la permission dfinie restreint celles affectes un utilisateur donn par des politiques dfinies pour d'autres profils. Voir Restriction d'accs [p 307].
Actions sur les jeux de donnes	Cette section spcifie les permissions des actions sur les jeux de donnes.
Crer un le jeu de donnes enfant	Indique si le profil peut crer un jeu de donnes enfants. L'hritage doit aussi tre activ dans le modle de donnes.
Dupliquer le jeu de donnes	Indique si le profil peut dupliquer le jeu de donnes.
Supprimer le jeu de donnes	Indique si le profil peut supprimer le jeu de donnes.
Activer/dsactiver le jeu de donnes	Indique si le profil peut modifier la proprit "Activ" dans les informations du jeu de donnes. Voir Dsactivation d'un jeu de donnes [p 139].
Crer une vue	Indique si le profil peut crer des vues et des hirarchies.
Droits sur tables	Spcifie les permissions par dfaut pour toutes les tables. Des permissions spcifiques peuvent tre appliques une ou plusieurs tables en cliquant sur le bouton '+' sous Droits spcifiques par table.
Droits par dfaut >Crer un nouvel enregistrement	Indique si le profil peut crer des enregistrements dans une table.
Droits par dfaut >Surcharger des enregistrements	Indique si le profil peut remplacer des enregistrements hrits dans une table. Cette permission est utile quand on utilise l'hritage de jeu de donnes.
Droits par dfaut >Occulter des enregistrements	Indique si le profil peut occulter des enregistrements hrits dans une table. Cette permission est utile quand on utilise l'hritage de jeu de donnes.
Droits par dfaut >Supprimer un enregistrement	Indique si le profil peut supprimer des enregistrements dans une table.

Droits sur valeurs	Spcifie les permissions d'accs par dfaut pour tous les lments (tables, groupes et champs) d'un jeu de donnes, et permet de dfinir des permissions pour des lments spcifiques. Les permissions d'accs par dfaut sont utilises, condition qu'il n'y ait pas de permission spcifique affecte un lment. Le slecteur de droits spcifiques permet d'attribuer des permissions d'accs spcifiques un lment. Les liens "Lecture", "Ecriture" et "Non visible" dterminent les permissions d'accs correspondant l'lment slectionn. Il est possible de retirer une permission d'accs spcifique en utilisant le lien "(par dfaut)".
Droits sur les services	Spcifie les permissions d'accs sur les services. Un service barr n'est pas accessible un profil.

CHAPITRE 23

Hritage entre jeux de donnes

En utilisant le concept d'hritage entre jeux de donnes, vous pouvez crer des jeux de donnes additionnels, partir d'un jeu de donnes racine. Ces jeux de donnes enfants hritent des proprits et des valeurs de leur parents, qui peuvent tre surcharges si ncessaire. Plusieurs niveaux d'hritage peuvent tre crs.

L'hritage peut tre utilis pour adapter des donnes de rfrence divers contextes. Par exemple, il serait possible de dfinir des valeurs globales dans un jeu de donnes parent, et de crer des jeux de donnes enfants par zones gographiques. Ceci permettra ces derniers d'hriter des valeurs de leur parent, et de les surcharger si besoin.

> **Note**
>
> Le comportement standard est d'interdire l'hritage de jeux de donnes. Il est donc ncessaire d'activer explicitement cette fonction au niveau du modle de donnes.

Voir aussi*Configuration du modle de donnes* *[p 43]*

Ce chapitre contient les sections suivantes :

1. Structure de l'hritage entre jeux de donnes
2. Hritage de valeurs

23.1 Structure de l'hritage entre jeux de donnes

Une fois le jeu de donnes racine cr, un jeu de donnes enfant peut tre cr l'aide du bouton **+**, situ dans l'cran de slection des jeux de donnes du panneau de navigation.

> **Note**
>
> - Un jeu de donnes ne peut pas tre supprim s'il a des jeux de donnes enfants. Ces enfants doivent tre supprims pralablement.
> - Si un jeu de donnes enfant est dupliqu, le jeu de donnes nouvellement cr sera insr dans l'arbre des jeux de donnes existants, au mme niveau de l'arbre que le jeu de donnes dupliqu.

23.2 Hritage de valeurs

Quand un jeu de donnes enfant est cr, il hrite de toutes les valeurs des champs et des enregistrements de tables de son parent. Un champ ou un enregistrement peut soit hriter ses valeurs, soit les surcharger.

Dans une vue tabulaire, les valeurs hrites sont signales par un repre dans le coin en haut gauche de la cellule.

Le bouton permet de surcharger une valeur.

Hritage d'enregistrement

Une table dans un jeu de donnes enfant hrite des enregistrements des tables de ses jeux de donnes anctres. La table dans le jeu de donnes enfant peut rajouter, diter ou supprimer des enregistrements. Des tats sont dfinis pour diffrencier les types d'enregistrement.

Racine	Un enregistrement racine est un enregistrement cr dans le jeu de donnes courant, qui n'existe pas dans les jeux de donnes anctres. Il sera hrit par les jeux de donnes enfants.
Hrit	Un enregistrement hrit est dfini dans un des jeux de donnes anctres du jeu de donnes courant.
Surcharg	Un enregistrement surcharg est un enregistrement hrit dont les valeurs sont dites dans le jeu de donnes courant. Les valeurs surcharges seront hrites par les jeux de donnes enfants.
Occult	Un enregistrement occult est un enregistrement hrit qui est supprim du jeu de donnes courant. Il apparaîtra toujours dans le jeu de donnes courant comme un enregistrement barr, mais il ne sera pas hrit par les jeux de donnes enfants.

Quand le bouton est activ, la valeur de l'enregistrement est hrite du jeu de donnes parent. Ce bouton peut tre dsactiv, afin de surcharger l'enregistrement ou la valeur. Pour un enregistrement occult, l'activation de ce bouton restaure l'tat hrit.

La table suivante rsume le comportement des enregistrements lorsque l'on cre, modifie, ou supprime un enregistrement, selon son tat initial.

Etat	Cration	Edition	Suppression
Racine	Cration normale d'un enregistrement. L'enregistrement nouvellement cr sera hrit par ses jeux de donnes enfant.	Edition normale d'un enregistrement. Les nouvelles valeurs seront hrites par les jeux de donnes enfants.	Suppression normale d'un enregistrement. L'enregistrement va disparaître du jeu de donnes courant ainsi que des jeux de donnes enfants.
Hrit	Si un enregistrement est cr l'aide de la cl primaire d'un enregistrement hrit existant, l'tat de l'enregistrement devient surcharg, et sa valeur sera celle soumise sa cration.	Un enregistrement hrit doit tre dclar comme surcharg pour que ses valeurs soient modifiables.	Supprimer un enregistrement hrit change son tat "occult".
Surcharg	Non applicable. Il est impossible de crer un nouvel enregistrement si la cl primaire est dj utilise.	Un enregistrement surcharg peut tre remis l'tat "hrit", mais sa valeur spcifique sera perdue. Les valeurs de l'enregistrement surcharg peuvent tre hrites ou modifies.	Supprimer un enregistrement surcharg change son tat "occult".
Occult	Si un enregistrement est cr en utilisant la cl primaire de l'enregistrement existant occult, l'tat de l'enregistrement devient "surcharg" et sa valeur sera celle soumise la cration.	Non applicable. Un enregistrement occult ne peut plus tre dit.	Non applicable. Un enregistrement occult est dj considr comme supprim, et ne peut donc pas tre supprim une deuxime fois.

Voir aussi*Mcanisme de rsolution d'enregistrement* *[p 296]*

Modèles de workflow

CHAPITRE **24**

Introduction aux modles de workflow

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Utilisation de l'interface de modlisation de workflow
3. Modles de message gnriques
4. Limitations de workflows

24.1 Prsentation

Dfinition d'un modle de workflow

Dans TIBCO EBX, les workflows facilitent la gestion collaborative des donnes dans le rfrentiel. Un workflow peut contenir des actions utilisateurs sur les donnes ainsi que des tches automatiques, tout en mettant des notifications sur diffrents vnements.

La premire tape pour raliser un workflow est de crer un *modle de workflow* qui dfinit la succession d'tapes, les implications des utilisateurs, ainsi que le comportement du workflow.

Une fois qu'un modle de workflow est dfini, il peut tre valid et publi comme *publication de workflow*. Ensuite, les workflows de donnes peuvent tre lancs partir de la publication de workflow pour excuter les tapes dfinies dans le modle de workflow.

Voir aussi

Modle de workflow (glossaire) *[p 29]*

Workflow de donnes (glossaire) *[p 31]*

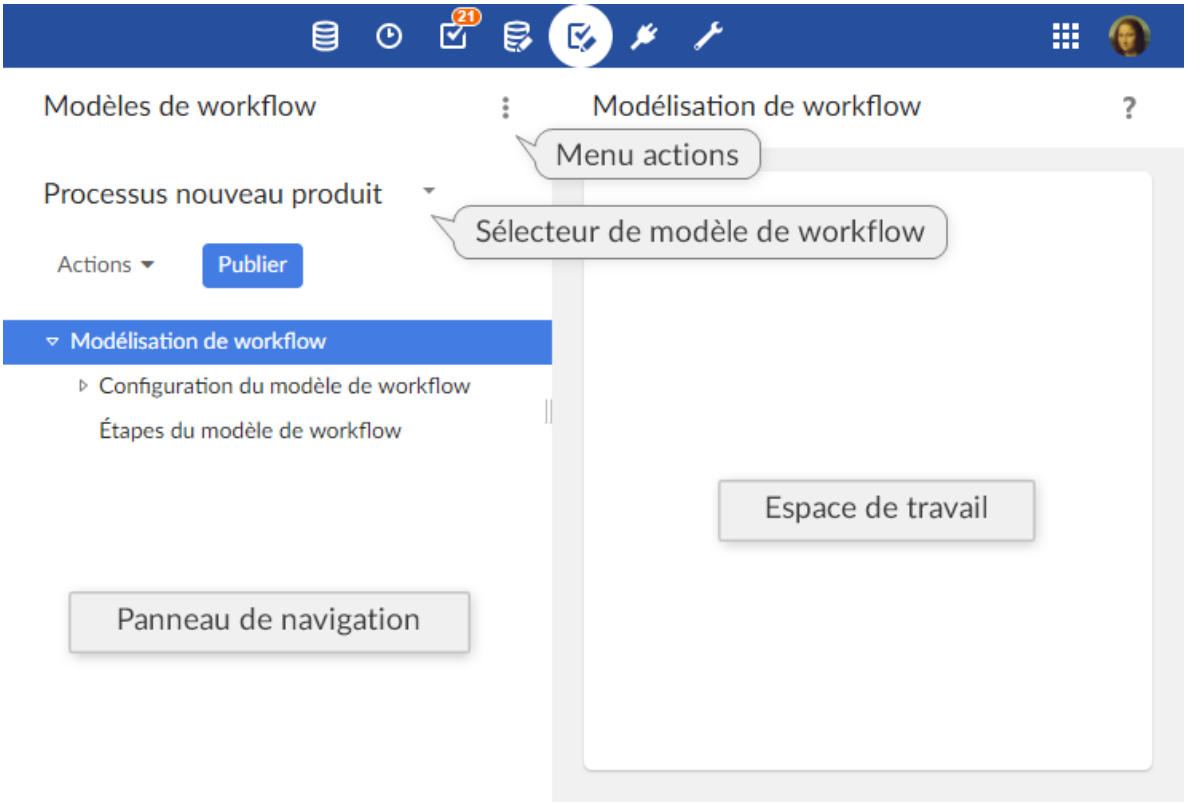
Concepts de base utiliss dans la modlisation des workflows

Une comprhension des termes suivants est ncessaire pour raliser la cration de modles de workflows :

- tche automatique [p 30]
- tche utilisateur [p 30]
- bon de travail [p 31]
- condition de workflow [p 30]
- appel des sous-workflows [p 30]
- tche d'attente [p 30]

- contexte des donnes [p 30]

24.2 Utilisation de l'interface de modlisation de workflow



> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance'. Seuls les utilisateurs autoriss peuvent accder ces interfaces spcifiques.

24.3 Modles de message gnriques

Des emails de notification peuvent tre envoys pour notifier les utilisateurs d'vnements spcifiques pendant l'excution d'un workflow.

Les modles de message peuvent tre dfinis et rutiliss dans n'importe quel modle de workflow dans le rfrentiel. Pour modifier les modles de message gnriques, slectionnez "Modles de message" dans le menu 'Actions' de la section Modles de Workflow.

Ces modles, qui sont partags par tous les modles de workflows, sont figs et inclus dans chaque publication de workflow. Ainsi, pour prendre en compte les modifications des modles de message, il sera ncessaire de mettre jour les publications existantes en re-publiant les modles de workflow concerns.

A noter galement que, lors d'un archivage, si l'on souhaite sauvegarder ces modles de messages, le jeu de donnes "configuration" doit tre slectionn car celui-ci contient ces modles.

A la cration d'un modle de message gnrique, deux champs sont obligatoires :

- 'Libell & Description':spcifie les libells et descriptions associs ce modle de message, localis.

- 'Message':spcifie l'objet de l'email et son corps de texte, localiss.

Le 'Type du message' est une donne facultative.

Le message peut inclure des variables du contexte de donnes sous la forme *${nom.variable}*, qui seront values lorsque le message sera envoy. De plus, les variables systme suivantes peuvent tre incluses:

system.time	Heure systme du rfrentiel.
system.date	Date systme du rfrentiel.
workflow.lastComment	Dernier commentaire sur la tche utilisateur prcdente. (Note:cette variable concerne la dernire tche utilisateur, et non la courante. Est considre courante la tche sur laquelle est positionn le workflow, incluant galement la notification de compltion de tche utilisateur).
workflow.lastDecision	Dernires dcisions sur la tche utilisateur prcdente. (Note:cette variable concerne la dernire tche utilisateur, et non la courante. Est considre courante la tche sur laquelle est positionn le workflow, incluant galement la notification de compltion de tche utilisateur).
user.fullName	Nom complet de l'utilisateur notifi.
user.login	Login de l'utilisateur notifi.
workflow.process.label	Libell du workflow en cours.
workflow.process.description	Description du workflow en cours.
workflow.workItem.label	Libell du bon de travail en cours.
workflow.workItem.description	Description du bon de travail en cours.
workflow.workItem.offeredTo	Rle auquel le bon de travail courant a t propos.
workflow.workItem.allocatedTo	Utilisateur, qui le bon de travail en cours a t allou.
workflow.workItem.link	Lien d'accs au bon de travail courant dans la corbeille, au moyen de l'API du composant web. Pour que ce lien soit calcul, il est ncessaire qu'un bon de travail courant soit dfini et que l'URL soit configure dans Workflow-executions, dans la configuration de mail.

workflow.workItem.link.allocateAndStart	Lien d'accs au bon de travail courant dans la corbeille, au moyen de l'API du composant web. Si le bon de travail cible n'est pas encore dmarr, il sera automatiquement allou l'utilisateur qui a cliqu sur le lien, puis dmarr. Pour que ce lien soit calcul, il est ncessaire qu'un bon de travail courant soit dfini et que l'URL soit configure dans Workflow-executions, dans la configuration de mail.
workflow.currentStep.label	Libell de l'tape courante.
workflow.currentStep.description	Description de l'tape courante.

Exemple

Modles de message gnrique :

```
Aujourd'hui  ${system.time}, un nouveau bon de travail vous a t propos.
```

Email rsultant :

```
Aujourd'hui  15:19, un nouveau bon de travail vous a t propos.
```

24.4 Limitations de workflows

Les fonctionnalits suivantes ne sont pas supportes :

- **Tches programmes**, tches excutes ds lors que leur tour vient, et dont l'excution ne peut pas tre reporte.
- **Tches vnementielles** permettant au workflow de progresser quand il reoit un vnement, du type appel web service.
- **Limitation temporelle** sur la dure d'une tche.

Concepts apparentés*Workflows de donnes* *[p 176]*

CHAPITRE **25**

Modlisation du workflow

Ce chapitre contient les sections suivantes :

1. Cration d'un modle de workflow
2. Implmentation des tapes
3. Tche utilisateur
4. Tche autonome
5. Conditions
6. Appels des sous-workflows
7. Tches d'attente
8. Visualisation du diagramme de workflow

25.1 Cration d'un modle de workflow

Un modle de workflow peut tre cr partir de la section 'Modle de workflow'. La seule information requise la cration est un nom de modle unique dans le rfrentiel.

Les tapes du modle de workflow sont initialises avec une transition initiale. Pour implmenter le modle de workflow, il faut dfinir la squence des tapes qui suivent cette transition initiale.

25.2 Implmentation des tapes

Un modle de workflow dfinit des tapes correspondant des oprations et des conditions. Les tapes possibles sont les suivantes :

- Tche utilisateur
- Tche automatique
- Condition
- Appel des sous-workflows
- Tche d'attente

Un contexte de donnes est li chaque workflow de donnes. Ce contexte de donnes peut tre utilis pour dfinir des variables, qui pourront tre utilises en entre et/ou en sortie dans les diffrentes tapes du workflow.

Stratgie d'avancement de l'tape suivante

Pour chaque type d'tape (except pour les appels de sous-workflows), une proprit est disponible pour prciser la stratgie d'avancement pour l'tape suivante. A la terminaison de l'tape, cette stratgie est value pour conditionner la navigation lors de l'excution du workflow. Par dfaut, la stratgie d'avancement est 'Afficher la table des bons de travail'. Dans ce cas, aprs l'excution de l'tape, la table des bons de travail est automatiquement affiche pour slectionner le prochain bon de travail ouvrir.

Une autre stratgie est disponible:'Ouvrir automatiquement l'tape suivante'. Cette stratgie permet l'utilisateur de garder la main sur ce workflow et d'excuter directement l'tape suivante. Si, la suite de cette excution, un bon de travail est atteint et que l'utilisateur connect peut le dmarrer, le bon de travail est automatiquement ouvert (si plusieurs bons de travail sont atteints, le premier cr est ouvert). Sinon, la stratgie d'avancement de l'tape suivante est value. Si aucun bon de travail n'est finalement atteint, la table des bons de travail est affiche.

Cette stratgie est prconise pour excuter plusieurs tapes la suite sans repasser par la corbeille de tches.

Il existe actuellement plusieurs limitations qui font que cette stratgie peut tre ignore. Dans ce cas, la table des bons de travail est automatiquement affiche. Les cas o cette proprit est ignore sont les suivants:si l'tape suivante est un sous-workflow, ou si l'tape courante est une tche utilisateur avec plus d'un bon de travail.

Dans le cas des conditions, deux autres stratgies sont proposes : 'Si vrai, ouvrir automatiquement l'tape suivante' et 'Si faux, ouvrir automatiquement l'tape suivante'. Ces stratgies permettent de conditionner la stratgie appliquer en fonction du rsultat de la condition.

Masque dans la vue graphique

Pour chaque type d'tat, une proprit est disponible pour dfinir quelles tapes doivent tre masques dans la vue graphique de workflow par dfaut.

Si cette proprit est active, l'tape sera automatiquement masque dans la vue graphique de workflow pour les utilisateurs non-administrateurs (ni administrateurs built-in, ni administrateurs de workflow).

25.3 **Tche utilisateur**

Une tche utilisateur est une tape qui ncessite une action de la part d'un utilisateur humain. Son libell et sa description peuvent tre localiss.

Mode

Pour des raisons de compatibilit ascendante, deux modes de tche utilisateur sont disponibles:le mode par dfaut et le mode de compatibilit.

Dans le mode par dfaut, une tche utilisateur gnre un seul bon de travail. Ce mode propose d'avantage de fonctionnalits, comme proposer un bon de travail une liste de profils, ou afficher les avatars directement dans la vue graphique de workflow.

Dans le mode de compatibilit, une tche utilisateur peut gnrer plusieurs bons de travail.

Par dfaut, le service de cration de tche utilisateur est masqu en mode compatiblit. Pour l'afficher, il faut activer une proprit dans le fichier `ebx.properties`. Pour plus d'informations, voir Disabling user task legacy mode [p 384].

Liste de profils

La dfinition des profils d'une tche utilisateur varie en fonction du mode de la tche utilisateur.

[Mode par dfaut] Propose aux profils suivants

Les profils dfinis sont les rles ou les utilisateurs auxquels la tche utilisateur est propose. Lors de l'excution de la tche utilisateur, un seul bon de travail est gnr. Si un seul utilisateur est dfini, le bon de travail est automatiquement allou cet utilisateur. Si un rle est dfini, le bon de travail est propos aux membres du rle. Si plusieurs utilisateurs et rles sont dfinis, le bon de travail est propos la fois ces utilisateurs et aux membres de ces rles.

[Mode de compatibilit] Participants

Les participants sont les rles ou les utilisateurs auxquels la tche utilisateur est destine. Par dfaut, lors de l'excution de la tche utilisateur, un bon de travail est gnr par profil. Si un profil est un utilisateur, le bon de travail est automatiquement allou cet utilisateur. Si un profil est un rle, le bon de travail est propos aux membres du rle.

For more information

Voir aussi*Extension* *[p 157]*

Service

TIBCO EBX propose les services prdfinis suivants :

- Accder des donnes
- Accder l'interface de fusion des espace de donnes
- Accder un espace de donnes
- Comparer deux contenus
- Crer un nouvel enregistrement
- Dupliquer un enregistrement
- Exporter les donnes depuis une table au format CSV
- Exporter les donnes depuis une table au format XML
- Fusionner un espace de donnes
- Importer les donnes dans une table depuis un fichier CSV
- Importer les donnes dans une table depuis un fichier XML
- Valider un espace de donnes, une image ou un jeu de donnes

Voir aussi*Services prdfinis de EBX* *[p 221]*

Configuration

Options gnrales > Rejet activ

Par dfaut, seulement l'action *accepter* est propose l'utilisateur lorsqu'il enregistre sa dcision.

Il est possible d'activer l'action 'rejeter' en positionnant ce champ 'Oui'.

Options gnrales > Demande de confirmation active

Par dfaut, quand un utilisateur enregistre sa dcision en cliquant sur le bouton 'Accepter' ou 'Rejeter', une demande de confirmation est affiche.

Il est possible de dsactiver cette demande de confirmation de la dcision en positionnant ce champ 'Non'.

Options gnrales > Activation des commentaires

Par dfaut, les commentaires sont activs. Lorsqu'un bon de travail est ouvert, un bouton 'Commentaires' est affich et permet l'utilisateur de saisir un commentaire.

Il est possible de masquer ce bouton 'Commentaires' en positionnant cette proprit *Non*.

Options gnrales > Caractre obligatoire du commentaire

Par dfaut, le commentaire associ un bon de travail est optionnel.

Il est possible de forcer l'utilisateur saisir un commentaire avant d'enregistrer sa dcision en positionnant ce champ sur le comportement attendu:'toujours obligatoire', 'obligatoire seulement si le bon de travail a t accept' ou 'obligatoire seulement si le bon de travail a t rejet'.

Options gnrales > Libells personnaliss

Durant l'excution des tches utilisateur, l'utilisateur peut accepter ou rejeter son bon de travail en cliquant sur le bouton correspondant. Dans la modlisation de workflow, il est possible pour certaines tches utilisateur de dfinir un libell et un message de confirmation personnaliss pour ces boutons. Cette fonctionnalit est particulirement utile pour ajouter une signification particulire l'action d'accepter ou rejeter un bon de travail.

[Mode de compatibilit] Terminaison > Critre de fin de tche

Une tche utilisateur peut tre assigne plusieurs *participants* et gnrer plusieurs bons de travail durant l'excution du workflow. Lors de la dfinition d'une tche utilisateur dans le modle de workflow, il est possible de slectionner un des critres prdfinis qui dterminent quand une tche utilisateur est termine, en se basant sur le statut des bons de travail associs. Lorsque la condition de sortie de la tche utilisateur sera remplie, le workflow de donnes avancera jusqu' l'tape suivante dfinie dans le modle.

Par exemple, pour le cas d'une tche utilisateur qui demande la validation de l'enregistrement d'un produit, il est possible de dsigner trois participants. Le critre de fin de tche permet de prciser si l'enregistrement du produit doit tre valid par les trois participants, ou uniquement par le premier utilisateur rpondre.

Le critre de fin de tche par dfaut est 'Quand tous les bons de travail ont t accepts'.

Remarque : Si une extension de service surcharge la mthode `UserTask.handleWorkItemCompletion` pour grer la fin de la tche utilisateur, c'est la charge du dveloppeur de l'extension de vrifier dans le code de cette mthode que le critre de fin de tche dfinie dans l'interface a t satisfait. Voir `UserTask.handleWorkItemCompletion`[API] dans la JavaDoc pour plus d'informations.

[Mode de compatibilit] Terminaison > Tolrance de rejet

Par dfaut, si un utilisateur rejette un bon de travail durant l'excution d'un workflow, la tche utilisateur est positionne en erreur et l'avancement du workflow est stopp.

Pour changer ce comportement par dfaut, il est possible de dfinir un nombre de bons de travail rejets tolrer. Tant que la limite de rejets tolrs n'est pas dpasse, aucune erreur ne se produit et c'est le critre de fin de tche qui dtermine quand la tche utilisateur est termine.

Les critres de fin de tche suivants tolrent automatiquement tous les rejets :

- 'Quand tous les bons de travail ont t accepts ou rejets'
- 'Quand tous les bons de travail ont t accepts, ou ds qu'un bon de travail a t rejet'

Extension

Il est possible d'tendre programmatiquement le comportement de la tche afin que cette dernire s'insre plus finement dans le contexte du workflow. Par exemple si un comportement spcifique est ncessaire, pour la cration du bon de travail ou pour terminer la tche de l'utilisateur.

La rgle spcifie est une classe Java Bean qui doit tendre la classe `UserTask`[API].

> **Attention**
> Si une rgle est spcifie et la mthode `handleWorkItemCompletion` surcharge, la stratgie de fin n'est plus automatiquement contrle. C'est au dveloppeur de la vrifier dans cette mthode.

Notification

Une notification par courrier lectronique peut tre envoye aux utilisateurs quand des vnements spcifiques se produisent. Pour chaque vnement, vous pouvez spcifier un message type utiliser.

En outre, il est possible de dfinir un profil, auquel une copie de chaque courrier envoy sera transmise.

Voir aussi*Modles de message gnriques* *[p 149]*

Relance

Des courriers lectroniques de relance pour les bons de travail proposs ou allous et inachevs peuvent tre envoys aux utilisateurs concerns de manire priodique.

Le contenu des courriers de relance dpend de l'tat courant du bon de travail. En effet, si le bon de travail est propos, la notification utilisera le modle de courrier lectronique 'Bons de travail proposs' ; si le bon de travail est allou, la notification utilisera le modle 'Bons de travail allous'.

Echance

Une tche utilisateur peut avoir une chance. Quand cette date est atteinte et si les bons de travail associs n'ont pas t termins, un courrier lectronique spcifique est envoy aux utilisateurs concerns. Ce courrier lectronique va alors tre renvoy tous les jours jusqu' l'achvement de la tche.

Il y a deux types d'chances:

- *Echance absolue :* une date du calendrier.
- *Echance relative :* dure (en heures, jours ou mois). La dure est value partir d'une date de rfrence:dbut d'une tche utilisateur ou dbut d'un workflow.

25.4 Tche autonome

Il existe deux types de tches automatiques :

Script de la bibliothque	EBX fournit des scripts de la bibliothque prdfinis, qui peuvent tre utiliss directement. Les scripts de la bibliothque spcifiques doivent tre dclars dans un fichier `module.xml`. Un script de la bibliothque spcifique doit dfinir son libell, sa description et ses paramtres. Quand un utilisateur slectionne un script de la bibliothque dans une tape d'un modle de donnes, les paramtres associs sont affichs dynamiquement.
Script spcifique	Spcifie une classe Java qui excute des actions spcifiques. La classe associe doit tre dans le mme module que celui associ au modle de workflow. Ses libells et ses descriptions ne sont pas affichs dynamiquement aux utilisateurs dans un modle de workflow.

Packaging TIBCO EBX modules [p 483]

Script de la bibliothque

EBX inclut les scripts de la bibliothque prdfinis suivants :

- Crer un espace de donnes
- Crer une image
- Fusionner un espace de donnes
- Importer une archive
- Fermer un espace de donnes
- Modifier une variable du contexte de donnes
- Envoyer un courrier lectronique
- Supprimer des enregistrements (Note:ce script peut supprimer plusieurs enregistrements la fois)

Les scripts de la bibliothque sont des classes qui tendent la classe `ScriptTaskBean`[API]. En complment des scripts de la bibliothque prdfinis, vous pouvez dfinir des scripts de la bibliothque additionnels pour les utiliser dans les modles de workflows. Leurs libells et descriptions peuvent tre localiss.

La mthode `ScriptTaskBean.executeScript`[API] est appele lorsque le workflow de donnes atteint l'tape correspondante.

Attention

La mthode `ScriptTaskBean.executeScript`[API] ne doit crer aucun `Thread`. En effet, le moteur de workflow passera l'tape suivante quand le script se terminera. L'excution d'une partie du script en mode asynchrone ne garantit donc pas sa terminaison avant le passage l'tape suivante.

Voir l'exemple [p 574].

Il est possible de paramtrer dynamiquement les variables du script de la bibliothque, si son implmentation suit la spcification Java Bean. Les variables doivent tre dclares comme paramtres du script de la bibliothque dans le fichier `module.xml`. Le contexte des donnes n'est pas accessible depuis le Java bean.

> **Note**
>
> Certains scripts de bibliothque prdfinis sont marqus comme "obsoltes" car ils ne sont pas compatibles avec internationalisation. Il est recommand d'utiliser les nouvelles tches qui sont compatibles avec internationalisation.

Scripts spcifiques

Les scripts spcifiques ont la particularit d'tendre la classe Sample of ScriptTask [p 574].

La mthode `ScriptTask.executeScript`[API] est appele lorsque le workflow de donnes atteint l'tape correspondante.

> **Attention**
>
> La mthode `ScriptTask.executeScript`[API] ne doit crer aucun `Thread`. En effet, le moteur de workflow passera l'tape suivante quand le script se terminera. L'excution d'une partie du script en mode asynchrone ne garantit donc pas sa terminaison avant le passage l'tape suivante.

Voir l'exemple [p 574].

Il n'est pas possible de paramtrer dynamiquement les variables d'un script spcifique. Toutefois, le contexte des donnes est accessible depuis le Java Bean.

25.5 Conditions

Les conditions sont des tapes dcisionnelles du workflow.

Il existe deux types de conditions qui, une fois dfinies, peuvent tre utilises dans les tapes de modlisation du workflow :

Condition de la bibliothque	EBX fournit des conditions de la bibliothque prdfinies, qui peuvent tre utilises directement. Les conditions de la bibliothque spcifiques doivent tre dclares dans un fichier `module.xml`. Une condition de la bibliothque spcifique doit dfinir son libell, sa description et ses paramtres. Quand un utilisateur slectionne une condition de la bibliothque dans une tape d'un modle de donnes, les paramtres associs sont affichs dynamiquement.
Condition spcifique	Spcifie une classe Java qui implmente une condition spcifique. La classe associe doit tre dans le mme module que celui associ au modle de workflow. Ses libells et ses descriptions ne sont pas affichs dynamiquement aux utilisateurs dans un modle de workflow.

Packaging TIBCO EBX modules [p 483]

Condition de la bibliothque

EBX inclut les conditions de la bibliothque prdfinies suivantes :

- Espace de donnes modifi ?
- Espace de donnes valide ?
- Dernire tche utilisateur accepte ?
- Valeur nulle ou vide ?
- Valeurs gales ?

Les conditions de la bibliothque sont des classes qui tendent la classe `ConditionBean`[API]. En complment aux conditions de la bibliothque prdfinies, vous pouvez dfinir des conditions de la bibliothque additionnelles pour les utiliser dans des modles de donnes. Leurs libells et descriptions peuvent tre localiss.

Voir l'exemple [p 577].

Il est possible de paramtrer dynamiquement les variables de la condition de la bibliothque, si son implmentation suit la spcification Java Bean. Les variables doivent tre dclares comme paramtres du script de la bibliothque dans le `module.xml`. Le contexte des donnes n'est pas accessible depuis le Java bean.

Conditions spcifiques

Les conditions spcifiques ont la particularit d'tendre la classe `Condition`[API].

Voir l'exemple [p 576] dans l'API Java.

Il n'est pas possible de paramtrer dynamiquement les variables d'une condition spcifique. Toutefois, le contexte des donnes est accessible depuis le Java bean.

25.6 **Appels des sous-workflows**

Les tapes du type appel des sous-workflows positionnent le workflow de donnes courant dans l'tat "en attente" et lancent un ou plusieurs sous-workflows.

Il est possible d'inclure un autre modle de workflow dans le modle de workflow courant en dfinissant un appel unique ce sous-workflow dans une tape.

Si plusieurs sous-workflows sont appels par une tape, ils sont excuts simultanment, en parallle. Tous les sous-workflows doivent tre termins pour que le workflow parent passe l'tape suivante. Le libell et la description de chaque sous-workflow peuvent tre localiss.

Deux types d'appels des sous-workflows existent :

Statique	Dfinit un ou plusieurs sous-workflows lancer chaque fois que cette tape est excute dans un workflow de donnes. Pour chaque sous-workflow, il est possible de spcifier ses libells et ses descriptions, ainsi que les mappings d'entre et de sortie dans son contexte de donnes. Ce mode est utile quand on sait l'avance quels sous-workflows doivent tre lancs, et comment le mapping de sortie doit tre realis.
Dynamique	Spcifie une classe Java qui implmente un appel spcifique des sous-workflows. Tous les workflows qui peuvent ventuellement tre appels comme sous-workflows par le code doivent tre dclars comme dpendances. Le contexte de donnes est directement accessible depuis le Java bean. Les appels de sous-workflows dynamiques doivent tre dclars dans un fichier `module.xml`. Ce mode est utile quand le lancement des sous-workflows est conditionnel (par exemple, lorsqu'il dpend d'une variable du contexte de donnes), ou quand le mapping de sortie dpend de l'excution des diffrents sous-workflows.

25.7 Tches d'attente

Une tape d'attente dans un modle de workflow met le workflow en pause en attendant la rception d'un vnement donn.

Lorsqu'une tche d'attente est appele, le moteur de workflow gnre un identifiant unique de rveil li la tche d'attente. Cet identifiant est requis pour rveiller la tche d'attente et, par consquent, le workflow associ.

Une tche d'attente spcifie quel profil est autoris rveiller la tche d'attente;et la classe Java qui implmente un bean de tche d'attente : `WaitTaskBean`[API].

Le contexte de donnes du workflow est directement accessible partir du Java bean.

Les beans de tche d'attente doivent tre dclars dans un fichier `module.xml`.

Le bean de tche d'attente est d'abord appel lorsque le workflow est mis en pause. L'identifiant de rveil est alors disponible pour appeler un web service, par exemple. Puis, le bean de tche d'attente est appel lorsque la tche d'attente est rveille. De cette faon, le contexte de donnes peut tre mis jour en fonction des paramtres reus.

> **Note**
>
> L'administrateur built-in est toujours autoris rveiller un workflow.

25.8 Visualisation du diagramme de workflow

A propos

Lorsqu'un modle de workflow est dfini, il est possible de le visualiser sous forme d'un diagramme qui se rapproche de la norme BPMN.

Le service est disponible via un bouton ddi dans la vue hirarchique. Le bouton est le suivant:

Ce service fournit une vue aux possibilits d'dition limites:il est possible d'diter les tapes existantes, mais pas d'en modifier les liens ni de crer de nouvelles tapes.

Notez galement que ce diagramme s'inspire des normes BPMN mais n'en est pas une stricte implmentation, puisque les concepts du workflow d'EBX sont lgrement diffrents.

Sauvegarde de la mise en page

Il est possible de sauvegarder la mise en page du diagramme et de le sauvegarder. Notez toutefois que la sauvegarde est globale et non locale;elle est donc partage par tous les utilisateurs.

Actions

Exporter en PNG	Cre une image PNG du modle.
Exporter en SVG	Cre une image SVG du modle.
Exporter en PDF	Crer un PDF sur une page

Vue

Mise en page > Mise en page par dfaut	Applique la mise en page par dfaut.
Grille > Afficher/Masquer la grille	Affiche la grille lorsque celle-ci est invisible, sinon masque la grille.

Boutons

Sauvegarder	Sauvegarde la mise en page.
Sauvegarder et fermer	Sauvegarde la mise en page et ferme le service.
Recharger	Annule les modifications de la mise en page en cours et recharge la mise en page prcdemment sauvegarde.
Fermer	Ferme le service.

Fonctionnalits additionnelles

La vue graphique offre galement des fonctionnalits additionnelles

Annuler l'action prcdente	CTRL + Z
Zoomer/Dezoomer	Bouton du milieu de la souris puis roulette / CTRL puis roulette
Slection multiple	liquer sur le noeud ou lien slectionner tout en appuyant sur la touche CTRL / Dessiner un rectangle de slection (Il faut attendre 1 seconde aprs le clic pour activer la zone de slection)
Personnaliser le trac des liens	Lorsqu'on clique sur un lien, on peut dplacer chaque segment grce aux carrs de slection qui apparaissent sur les coins de chaque arte, ou fractionner le segment en dplaant le cercle qui apparaît au milieu de chaque segment
Aperu	Un panneau est maintenant disponible avec un diagramme de workflow miniaturis qui pourra tre utilis pour la navigation. Il sera possible de le rduire, l'agrandir et le dplacer dans la surface correspondant la vue du diagramme de workflow.

CHAPITRE **26**

Configuration du modle de workflow

Ce chapitre contient les sections suivantes :

1. Informations
2. Proprits du modle de workflow
3. Contexte de donnes
4. Personnalisation des vues d'excution de workflow
5. Permissions sur les workflows de donnes associs
6. Historique des images du modle de workflow
7. Suppression d'un modle de workflow

26.1 Informations

Pour visualiser et diter les informations concernant le propritaire et la documentation du modle de workflow, slectionnez "Informations" dans le menu "Actions" du modle de workflow [p 149] dans le panneau de navigation.

Propritaire	Personne pouvant diter les informations du modle de workflow et dfinir des rgles de permission.
Documentation locale	Libell et description associs au modle de workflow localiss.
Activation	*Cette proprit est obsolte.* Spcifie si un modle de workflow est activ. Un modle de workflow doit tre activ pour pouvoir tre publi.

26.2 Proprits du modle de workflow

La configuration d'un modle de workflow est accessible depuis le panneau de navigation.

Module	Module contenant les ressources Java spcifiques (extension de tche utilisateur, script et conditions spcifiques).
Notification de dmarrage	Liste des profils qui doivent recevoir une notification (choisie dans la liste des modles de messages) quand un workflow dmarre. Voir Modles de message gnriques [p 149].
Notification de fin	Liste des profils qui doivent recevoir une notification (choisie dans la liste des modles de messages), quand un workflow se termine. La notification n'est envoye que si le workflow a t termin "normalement" (pas par une action d'administration). Voir Modles de message gnriques [p 149].
Notification d'erreur	Liste des profils qui recevront une notification (choisie dans la liste des modles de messages) quand un workflow est mis en d'erreur. Voir Modles de message gnriques [p 149].
Priorit	Par dfaut, chaque workflow associ ce modle sera lanc avec cette priorit. Cette valeur est facultative. Si aucune valeur n'est dfinie ici, et une priorit par dfaut est dfinie pour le rfrentiel, la prioriti par dfaut pour le rfrentiel sera applique tous les workflows et bons de travail sans priorit. Voir Priorit de bons de travail [p 189] pour plus d'informations. **Note :** Seuls les utilisateurs dfinis en tant qu'administrateurs de workflows seront autoriss modifier la priorit des workflows de donnes associs manuellement.
Activer le lancement direct	Par dfaut, lorsqu'un workflow est lanc, il est propos l'utilisateur de saisir une documentation pour le nouveau workflow dans un formulaire intermdiaire. Cette documentation est facultative. Positionner la proprit "Activer le lancement direct" "Oui" permet d'viter cette tape de documentation et de lancer directement le workflow.
Ouvrir automatiquement la premire tape	Permet de conditionner la navigation, suite un lancement de workflow. Par dfaut, une fois le workflow lanc, la

	table courante (lanceurs de workflow ou monitoring > publications) est automatiquement affiche. Activer cette proprit permet au crateur du workflow de garder la main sur le workflow lanc. Si, suite l'excution de la premire tape du workflow, un bon de travail est atteint, et que ce bon de travail peut tre dmarr par le lanceur de workflow, le bon de travail est automatiquement ouvert (si plusieurs bons de travail sont atteints, le premier cr est ouvert). Cela vite au lanceur de slectionner le bon de travail correspondant dans la corbeille de tches. Si aucun bon de travail n'est atteint, la stratgie d'avancement de l'tape suivante est value. Si aucun bon de travail n'est finalement ouvert, la table partir de laquelle le workflow a t lanc est affiche. **Limitation :** Cette proprit est ignore si la premire tape est un appel de sous-workflow.
Trigger de workflowr	Composant interceptant les vnments principaux d'un workflow. Ce bean doit tre dclar dans un fichier `module.xml`. Voir l'exemple [p 580].
Permissions	Dfinit les permissions pour les actions lies au workflows de donnes associs au modle de workflow. Ce bean doit tre dclar dans un fichier `module.xml`. Voir l'exemple [p 579].
Permissions programmatiques	Dfinit le composant grant les permissions du workflow. S'il est dfini, il remplace l'ensemble des permissions dfinies dans la proprit 'Permissions'.

26.3 Contexte de donnes

La configuration du contexte de donnes est accessible depuis le panneau de navigation.

Chaque workflow possde un contexte de donnes qui lui est propre, lui permettant ainsi de disposer d'un espace de donnes local durant son excution. Cela permet de stocker et faire varier des valeurs qui orienteront l'excution du workflow.

Le contexte de donnes est dfini par une liste de variables. Chaque variable possde les proprits suivantes :

Nom	Identifiant de la variable.
Valeur par dfaut	Si dfinie, la variable sera initialise avec cette valeur par dfaut.
Paramtre d'entre	'Oui' doit tre coch pour dfinir cette variable comme un paramtre d'entre.
Paramtre de sortie	'Oui' doit tre coch pour dfinir cette variable comme un paramtre de sortie. Sinon, la variable ne sera pas affiche dans la liste des paramtres de sortie, dans l'interface de dfinition d'une tche.

26.4 **Personnalisation des vues d'excution de workflow**

La personnalisation des vues d'excution de workflow est accessible depuis le panneau de navigation.

Cette personnalisation permet de configurer les colonnes spcifiques des vues de bons de travail et de workflows (corbeille, monitoring des bons de travail, monitoring des workflows actifs et workflows termins). Pour chaque colonne spcifique, il est possible d'associer une expression qui peut contenir des variables du contexte de donnes qui seront values lors de l'affichage du workflow.

26.5 Permissions sur les workflows de donnes associs

Administration de workflow	Dfinit le profil autoris excuter des tches d'administration sur les workflows. Les actions d'administration sont : rejouer une tape, rveiller un workflow, terminer un workflow, dsactiver une publication et dpublier. Pour pouvoir excuter ces actions, ce profil bnficie automatiquement de la permission "Visualiser les workflows". L'administrateur built-in est toujours autoris administrer les workflows.
Administration de workflow > Rejouer une tape	Dfinit le profil autoris rejouer une tape de workflow. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour rejouer une tape, un bouton est disponible dans la section "Monitoring > Workflows actifs". Un profil qui a la permission "Administration de workflow" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris rejouer une tape de workflow.
Administration de workflow > Terminer un workflow	Dfinit le profil autoris terminer et supprimer un workflow. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour terminer et supprimer un workflow en cours, un bouton est disponible dans la section "Monitoring > Workflows actifs". Pour supprimer un workflow termin, un bouton est disponible dans la section "Workflows termins". Un profil qui a la permission "Administration de workflow" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris terminer un workflow.
Administration de workflow > Forcer le rveil d'un workflow	Dfinit le profil autoris forcer le rveil d'un workflow en attente. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour rveiller un workflow, un bouton est disponible dans la section "Monitoring > Workflows actifs". Un profil qui a la permission "Administration de workflow" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris rveiller un workflow.
Administration de workflow > Dsactiver une publication	Dfinit le profil autoris dsactiver une publication de workflow. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour dsactiver, un bouton est disponible dans la section "Monitoring > Publications" uniquement pour

	les publications actives. Un profil qui a la permission "Administration de workflow" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris dsactiver une publication.
Administration de workflow > Dpublier	Dfinit le profil autoris dpublier une publication de workflow. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour dpublier, un bouton est disponible dans la section "Monitoring > Publications" pour les publications dsactives uniquement. Un profil qui a la permission "Administration de workflow" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris dpublier.
Gestion de l'allocation	Dfinit le profil autoris grer l'allocation des bons de travail. Les actions d'allocation sont : allouer les bons de travail, rallouer les bons de travail et dsallouer les bons de travail. Pour pouvoir excuter ces actions, ce profil bnficie automatiquement de la permission "Visualiser les workflows". L'administrateur built-in est toujours autoris grer l'allocation des workflows.
Gestion de l'allocation > Allouer les bons de travail	Dfinit le profil autoris allouer les bons de travail. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour allouer un bon de travail, un bouton est disponible dans la section "Monitoring > Bons de travail" uniquement pour les bons de travail proposs. Un profil qui a la permission "Gestion de l'allocation" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris allouer les bons de travail.
Gestion de l'allocation > Rallouer les bons de travail	Dfinit le profil autoris rallouer les bons de travail. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour rallouer, un bouton est disponible dans la section "Monitoring > Bons de travail" uniquement pour les bons de travail allous. Un profil qui a la permission "Gestion de l'allocation" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris rallouer les bons de travail.
Gestion de l'allocation > Dsallouer les bons de travail	Dfinit le profil autoris dsallouer les bons de travail. Pour pouvoir excuter cette action, ce profil bnficie automatiquement de la permission "Visualiser les workflows". Pour dsallouer, un bouton est disponible dans la section "Monitoring > Bons de travail" uniquement pour les bons de travail allous. Un profil qui a la

	permission "Gestion de l'allocation" est automatiquement autoris effectuer cette action spcifique. L'administrateur built-in est toujours autoris dsallouer les bons de travail.
Lancer les workflows	Dfinit le profil autoris lancer manuellement de nouveaux workflows. Cette permission permet de lancer des workflows partir des publications actives dans la section "Lanceurs de workflow". L'administrateur built-in est toujours autoris lancer les workflows.
Visualiser les workflows	Dfinit le profil autoris visualiser les workflows. Par dfaut, un utilisateur final peut uniquement voir les bons de travail qui lui sont proposs ou allous dans la section "Boîte de rception". Cette permission permet en plus de visualiser les publications, workflows et bons de travail associs ce modle de workflow dans les sections "Monitoring" et "Workflows termins". Ce profil bnficie automatiquement de la permission "Visualiser les workflows termins". L'administrateur built-in est toujours autoris visualiser les workflows.
Visualiser les workflows > Le crateur d'un workflow peut le visualiser	Si activ, le crateur d'un workflow a la permission de visualiser les workflows qu'il a lancs. Cette permission restreinte lui donne accs aux workflows qu'il a lancs et aux bons de travail associs dans les sections "Monitoring > Workflows actifs", "Monitoring > Bons de travail" et "Workflows termins". La valeur par dfaut est "Non".
Visualiser les workflows > Visualiser les workflows termins	Dfinit le profil autoris visualiser les workflows termins. Cette permission permet de visualiser les workflows termins dans la section "Workflows termins" et d'accder leur historique. Un profil qui a la permission "Visualiser les workflows" est automatiquement autoris effectuer cette action. L'administrateur built-in est toujours autoris visualiser les workflows termins.

> **Note**
>
> Un utilisateur n'ayant aucun privilge particulier pourra voir un bon de travail uniquement si celui-ci lui est propos ou personnellement allou.

Voir aussi*Administration d'un workflow* *[p 193]*

26.6 Historique des images du modle de workflow

L'historique des images d'un modle de workflow peut tre gr sous **Actions > Historique**.

La table d'historique affiche toutes les images contenant le modle de workflow et indique si le modle a t publi. Pour chaque image, il est possible d'exporter ou d'afficher le modle de workflow correspondant en utilisant le bouton **Actions**.

26.7 Suppression d'un modle de workflow

Un modle de workflow peut tre supprim. Cependant, toute version publie auparavant reste accessible dans la section Workflows de Donnes. D'autre part, si un modle de workflow est recr avec un nom identique, lors d'une publication, un message demandera la confirmation de remplacement de la publication prcdente.

Voir aussi*Publication d'un modle de workflow* *[p 173]*

CHAPITRE **27**

Publication d'un modle de workflow

Ce chapitre contient les sections suivantes :

1. A propos des publications de workflow
2. Publication et images de modle de donnes
3. Sous-workflows dans les publications

27.1 A propos des publications de workflow

Ds qu'un modle de workflow est dfini, il doit tre publi afin d'autoriser les utilisateurs lancer des workflows de donnes associs. Une publication est ralise en cliquant sur le bouton 'Publier' dans le panneau de navigation.

Si aucune tape d'appel des sous-workflows n'est incluse dans le modle courant, l'option de publier d'autres modles en mme temps vous sera propose sur la page de publication. Si le modle de workflow actuel contient des tapes d'appel des sous-workflows, il doit tre publi seul.

Les modles de workflow peuvent tre publis plusieurs fois. Une publication est identifie par son nom de publication.

27.2 Publication et images de modle de donnes

Durant la publication d'un modle de workflow, une image est enregistre de son tat actuel. Vous pouvez prciser un libell et une description de l'image. Le libell par dfaut pour l'image est l'heure et la date au moment de publication. La description par dfaut indique l'utilisateur qui a publi le modle de workflow.

Pour chaque modle de workflow publi, le nom de la publication doit tre unique. Si un modle de workflow a dj t publi, il est possible de mettre jour une publication existante en rutilisant le mme nom de publication. Les noms des publications de workflow existantes sont proposs dans un menu. Dans le cas d'une mise jour de publication, l'ancienne version ne sera plus disponible pour lancer des workflows de donnes;mais elle permettra aux workflows existants de finir de s'excuter. Le contenu des diffrentes images peut tre consult dans l'historique des images de modle de workflow.

Voir aussi*Historique des images du modle de workflow* *[p 171]*

27.3 Sous-workflows dans les publications

Quand un modle de workflow, contenant un appel des sous-workflows, est publi, il n'est pas ncessaire de publier sparment les sous-workflows appels. D'un point de vue d'administration, le modle du

workflow principal (celui actuellement publi par l'utilisateur) et les modles de sous-workflows sont publis comme une entit unique.

Le systme tablit les dpendances avec les modles de workflows utiliss comme sous-workflows et cre automatiquement une publication par modle dpendant. Ces publications techniques sont exclusivement ddies au moteur de workflow pour le lancement des sous-workflows et ne sont donc pas disponibles dans la section Workflows de donnes.

La publication multiple n'est pas disponible pour un modle de workflow contenant un appel des sous-workflows. C'est pourquoi la premire tape de publication (slection des modles de workflow publier) n'est pas propose dans ce cas.

La republication du modle de workflow principal met automatiquement jour les modles de sous-workflows appels.

Bien qu'un sous-workflow puisse tre publi sparment comme modle de workflow principal, cela ne modifiera pas la version utilise par un autre modle de workflow principal publi qui utilise ce sous-workflow.

Workflows de données

CHAPITRE **28**

Introduction aux workflows de donnes

Ce chapitre contient les sections suivantes :

1. Prsentation

28.1 Prsentation

Un workflow de donnes est un processus excut sous la forme d'une succession d'tapes, dfinie par une publication de modle de workflow. Le workflow de donnes permet aux utilisateurs, ainsi qu'aux procdures automatises, d'effectuer des actions de faon collaborative sur les donnes. Une fois le modle de workflow spcifi et publi, la publication rsultante peut tre utilise pour lancer un workflow de donnes afin d'excuter les tapes dfinies.

En fonction des permissions dfinies par le modle de workflow, un utilisateur peut effectuer une ou plusieurs des actions suivantes sur les workflows de donnes associs :

- en tant qu'utilisateur avec les permissions par dfaut, effectuer les actions attendues par les bons de travail qui lui sont destins,
- en tant qu'utilisateur avec les permissions pour lancer les workflows, crer les nouveaux workflows de donnes depuis une publication du modle de workflow,
- en tant qu'utilisateur avec les permissions de monitoring de workflow, suivre l'avancement des workflows de donnes en cours, et consulter l'historique des workflows de donnes termins.
- en tant que gestionnaire d'allocation des bons de travail, modifier les allocations des bons de travail des autres utilisateurs manuellement.
- en tant qu'administrateur de workflow, effectuer diffrentes actions d'administration, comme par exemple, redmarrer une tape, terminer un workflow en cours ou rendre une publication indisponible pour le lancement de workflows.

Voir aussi

Bons de travail *[p 185]*

Lancement et monitoring de workflows de donnes *[p 191]*

Administration de workflows de donnes *[p 193]*

Permissions sur les workflows de donnes associs *[p 169]*

Concepts apparentés*Modles de workflow* *[p 148]*

CHAPITRE **29**

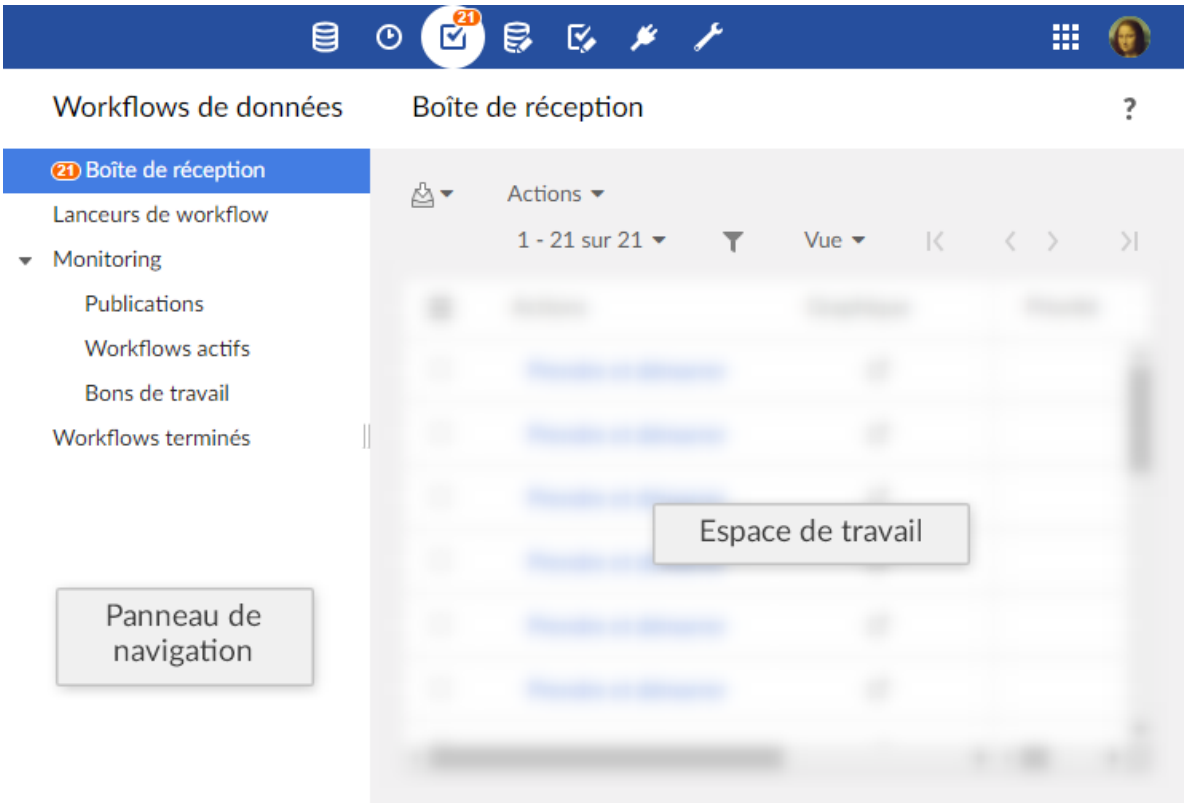
Utilisation de l'interface utilisateur de la section Workflow de donnes

Ce chapitre contient les sections suivantes :

1. Navigation dans l'interface utilisateur
2. Rgles de navigation
3. Vues personnalises
4. Colonnes spcifiques
5. Filtrage d'items dans les vues
6. Vue graphique d'un workflow de donnes

29.1 Navigation dans l'interface utilisateur

La fonctionnalit workflows de donnes se trouve dans la section **Workflows de donnes** dans l'interface utilisateur de TIBCO EBX.



> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cet cran via la 'Perspective avance' ou via une perspective spcifique. Seuls les utilisateurs autoriss peuvent accder ces interfaces spcifiques.

Le panneau de navigation est compos de plusieurs items. Ces items sont visibles en fonction des permissions globales associes. Les diffrents items sont :

Boîte de rception des bons de travail	Tous les bons de travail qui vous sont allous ou proposs, pour lesquels vous devez effectuer les actions spcifies.
Lanceurs de workflow	Liste des publications de workflow partir desquelles vous pouvez lancer des workflows de donnes, en fonction de vos permissions.
Monitoring	Vues pour le monitoring des workflows de donnes en cours pour lesquels vous avez les permissions ncessaires de visualisation.
Publications	Publications pour lesquelles vous avez les permissions ncessaires de visualisation. Si vous disposez de permissions d'administration supplmentaires, vous pouvez galement dsactiver la possibilit de lancer les workflows de donnes depuis une publication partir de cette vue.
Workflows actifs	Workflows de donnes, en cours d'excution, pour lesquels vous avez les permissions ncessaires de visualisation. Si vous disposez de permissions d'administration supplmentaires, vous pouvez galement effectuer des actions d'administration partir de cette vue, comme par exemple redmarrer une tape, ou terminer un workflow en cours.
Bons de travail	Bons de travail pour lesquels vous avez les permissions ncessaires de visualisation. Si vous disposez de permissions d'administration supplmentaires, vous pouvez galement effectuer des actions d'administration partir de cette vue, comme par exemple allouer les bons de travail aux utilisateurs ou rles.
Workflows termins	Workflows de donnes, dont l'excution est termine, pour lesquels vous avez les permissions ncessaires de visualisation. Si vous disposez de permissions d'administration supplmentaires, vous pouvez galement nettoyer les workflows termins partir de cette vue.

Note

Chaque section est accessible par composant web, par exemple pour intgration avec un portail, ou programmatiquement par la classe `ServiceKey` dans l'API Java.

Voir aussi

Utiliser TIBCO EBX comme composant web [p 211]

`ServiceKey`[API]

29.2 Rgles de navigation

Corbeille de bons de travail

Par dfaut, une fois qu'un bon de travail est excut, la corbeille de bons de travail est affiche.

Ce comportement peut tre modifi en fonction de la stratgie d'avancement de l'tape suivante. Cette stratgie permet d'enchaîner plusieurs tapes la suite sans repasser par la corbeille de tches.

Voir la stratgie d'avancement d'une tape de workflow [p 154] dans la modlisation de workflow.

Lanceurs de workflow

Par dfaut, une fois qu'un workflow est lanc, la table des lanceurs de workflow est affiche.

Ce comportement est modifiable dans la configuration du modle:il est possible d'ouvrir directement la premire tape sans afficher la table des lanceurs de workflow.

Voir l'ouverture automatique de la premire tape d'un workflow [p 166] dans la modlisation de workflow.

29.3 Vues personnalises

Il est possible de dfinir des vues sur les diffrentes tables du workflow et de bnficier de tous les mcanismes associs (dont la publication).

Les permissions pour crer et grer les vues des tables de workflows sont les mmes que les permissions des vues de tables de donnes. Il peut s'avrer ncessaire de modifier les permissions dans la section 'Administration' pour bnficier de cette fonctionnalit, en slectionnant *Gestion des workflows* > *Workflows*.

Voir les Vues [p 127] pour plus d'informations.

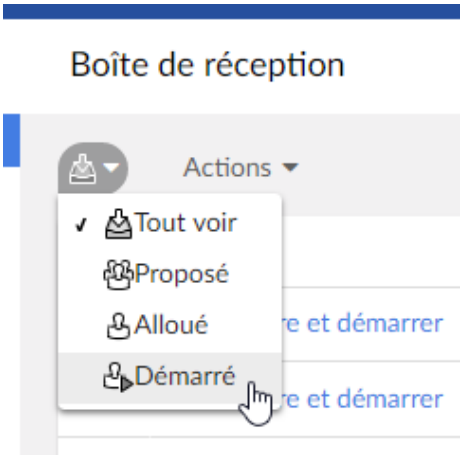
29.4 Colonnes spcifiques

Par dfaut, les colonnes spcifiques sont masques dans les vues qui peuvent en bnficier (corbeille, monitoring des bons de travail, monitoring des workflows actifs et workflows termins).

Il faut crer une vue personnalise et l'appliquer pour afficher les colonnes spcifiques souhaites. Pour chaque bon de travail ou workflow, la correspondance dfinie dans le modle de workflow associ est alors applique. Si une expression est dfinie pour une colonne et contient des variables du contexte de donnes, ces variables sont values lors de son affichage. Si une expression contient des expressions natives qui dpendent de la locale, l'expression est value dans la locale par dfaut.

29.5 **Filtrage d'items dans les vues**

Dans certaines vues, comme la 'Boîte de rception' des bons de travail, vous pouvez filtrer les lignes affiches dans les tables en fonction de leur tat. Dans ces vues, un menu est disponible cet effet pour slectionner l'tat correspondant au filtre attendu.



29.6 **Vue graphique d'un workflow de donnes**

En tant qu'utilisateur avec un bon de travail effectuer, ou en tant que superviseur ou administrateur de workflow de donnes, vous pouvez suivre l'avancement ou l'historique d'excution d'un workflow de donnes. Pour cela, cliquer sur le bouton 'Afficher' dans la colonne 'Workflow de donnes' des tables de l'interface de workflows de donnes. Ce bouton ouvre une pop-up qui affiche une vue interactive graphique de l'excution du workflow de donnes. Dans cette vue, vous pouvez suivre l'avancement global de l'excution, et slectionner une tape individuelle afin de consulter le dtail de ses informations.

Si des tapes ont t dfinies comme masques dans la modlisation de workflow, elles sont automatiquement masques dans la vue graphique de workflow pour les utilisateurs non-administrateurs (non-administrateurs built-in et non-administrateurs de workflow). Un bouton est disponible pour afficher les tapes masques. Le choix des utilisateurs (montrer ou masquer les tapes) est sauvegard par utilisateur, par publication, pendant la session utilisateur.

Pour les tches utilisateur dans le nouveau mode (un seul bon de travail), les informations principales propos du bon de travail unique sont directement affiches dans la vue graphique de workflow, si elles sont disponibles: l'avatar de l'utilisateur associ au bon de travail et la dcison qui a t prise pour le bon de travail (accept ou rejet).

CHAPITRE 30

Bons de travail

Ce chapitre contient les sections suivantes :

1. Prsentation des bons de travail
2. Travail sur un bon de travail en tant que participant
3. Priorit de bons de travail

30.1 Prsentation des bons de travail

Un bon de travail est une tche utilisateur unitaire qui doit tre effectue par un utilisateur humain. Par dfaut, quand un modle de workflow dfinit une tche utilisateur, les workflows de donnes, lancs depuis les publications du modle, gnrent un bon de travail individuel pour chacun des participants lists dans la tche utilisateur.

Voir aussi*Overview* *[p 587]*

tats d'un bon de travail

Lorsqu'un bon de travail est mis, pendant l'excution d'un workflow de donnes, pour une tche utilisateur dfinie dans le modle, le bon de travail peut prendre plusieurs tats:propos, allou, dmarr, et termin.

Cration des bons de travail

Mode par dfaut

Dans le mode par dfaut, un seul bon de travail est gnr quelle que soit la liste de profils dfinis.

Par dfaut, si un seul utilisateur est dfini dans la liste des profils, le bon de travail cr est l'tat *allou*.

Par dfaut, dans les autres cas, le bon de travail cr est l'tat *propos*.

> **Note**
>
> Le comportement par dfaut, dcrit ci-dessus, peut tre surcharg par une extension programmatique dfinie dans la tche utilisateur. Dans ce cas, les bons de travail peuvent tre gnrs programmatiquement, sans se baser obligatoirement sur la liste des profils de la tche utilisateur.

Mode de compatibilit

Par dfaut, pour chaque utilisateur dfini comme participant de la tche utilisateur, le workflow de donnes cre un bon de travail l'tat *allou*.

Par dfaut, pour chaque rle dfini comme participant de la tche utilisateur, le workflow de donnes cre un bon de travail l'tat *propos*.

> **Note**
>
> Le comportement par dfaut, dcrit ci-dessus, peut tre surcharg par une extension programmatique dfinie dans la tche utilisateur. Dans ce cas, les bons de travail peuvent tre gnrs programmatiquement, sans se baser obligatoirement sur la liste des participants de la tche utilisateur.

Variations des tats des bons de travail

Lorsque le bon de travail est l'tat *allou*, l'utilisateur dfini peut directement commencer travailler sur le bon de travail allou en effectuant l'action 'Prendre et dmarrer'. Le bon de travail passe alors l'tat *dmarr*.

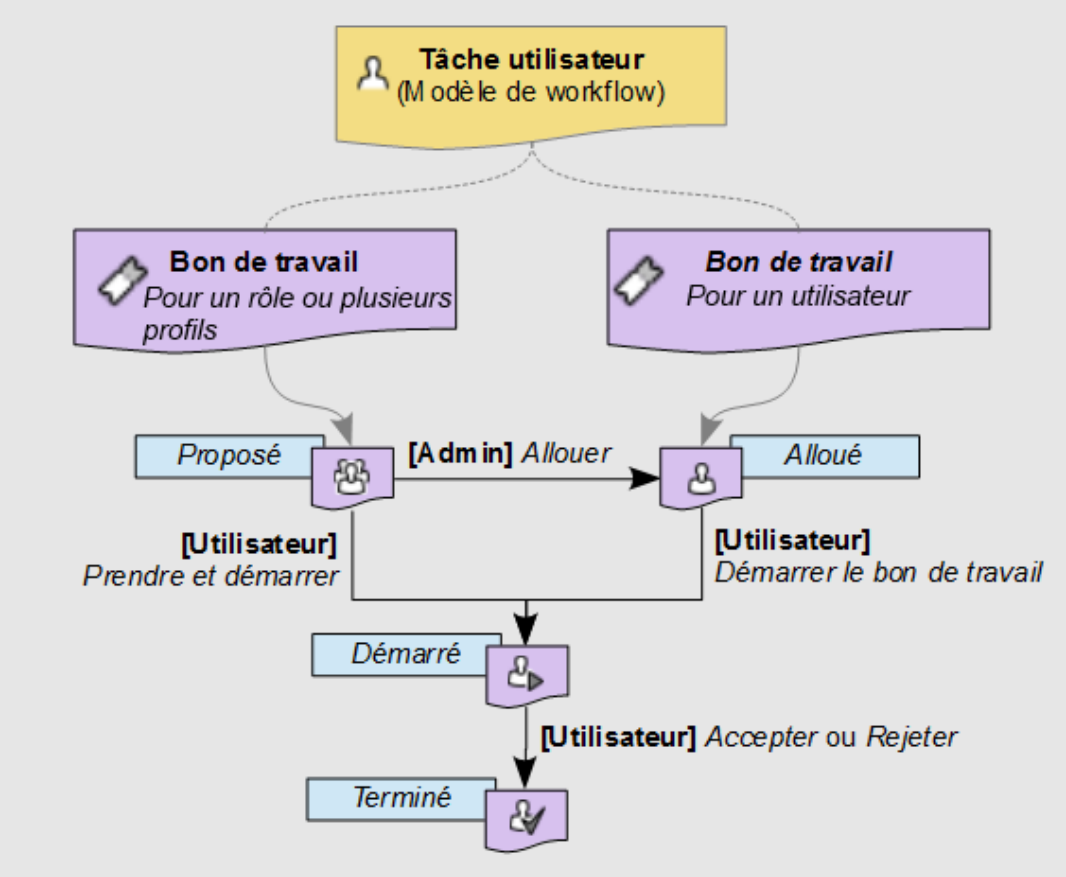
Lorsque le bon de travail est l'tat *propos*, tout utilisateur ou tout membre des rles auxquels il est propos peut prendre le bon de travail en utilisant l'action 'Prendre et dmarrer'. Le bon de travail passe ainsi l'tat *dmarr*.

Avant qu'un utilisateur ait pris le bon de travail propos, un gestionnaire des allocations du workflow de donnes peut intervenir pour affecter manuellement le bon de travail un utilisateur spcifique, passant ainsi le bon de travail l'tat *allou*. Puis, lorsque l'utilisateur commence travailler sur le bon de travail via l'action 'Dmarrer le bon de travail', le bon de travail passe l'tat *dmarr*.

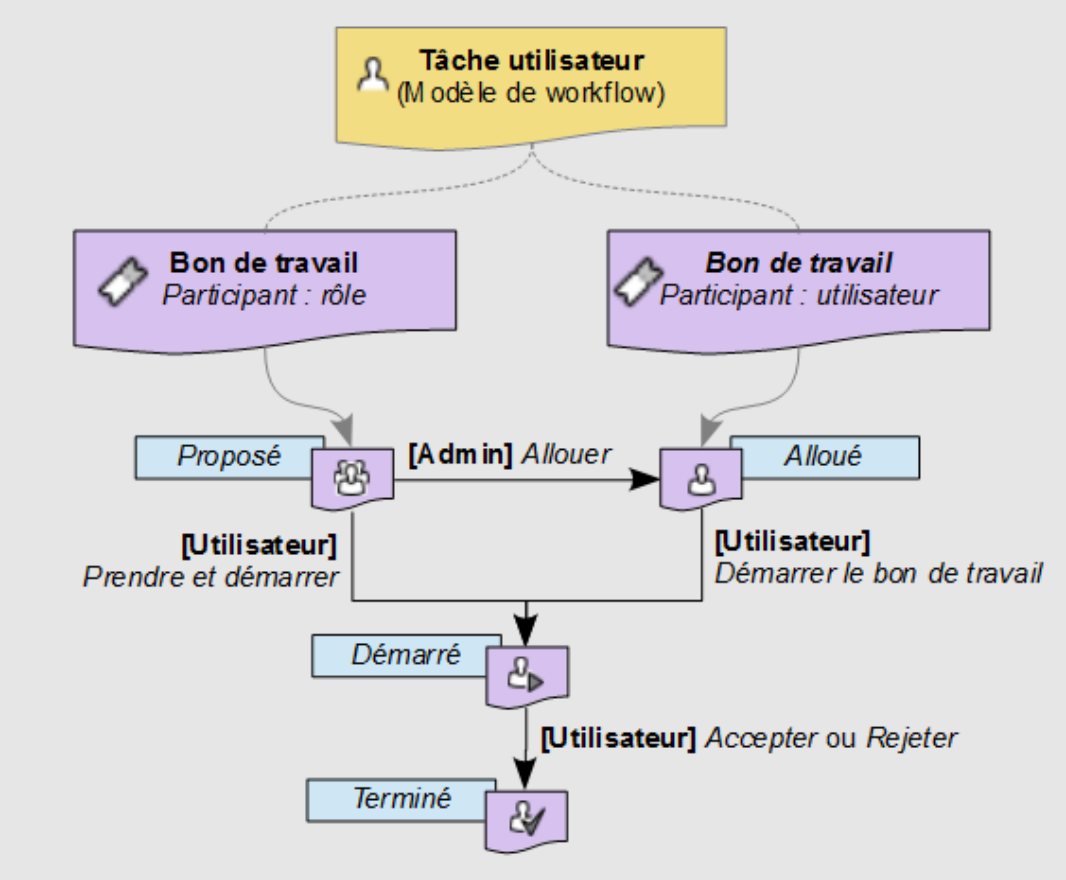
Une fois que l'utilisateur, qui a dmarr le bon de travail, a ralis l'action demande, l'action terminale 'Accepter' ou 'Rejeter' place le bon de travail dans l'tat *termin*. Lorsqu'un utilisateur termine un bon de travail, le workflow de donnes passe automatiquement l'tape suivante dfinie dans le modle de workflow.

Diagramme des tats de bon de travail

Mode par dfaut



Mode de compatibilit



30.2 Travail sur un bon de travail en tant que participant

Tous les bons de travail disponibles (qui vous sont soit proposs, soit allous), sont affichs dans votre boîte de rception des bons de travail. Quand vous commencez travailler sur un bon de travail, vous pouvez ajouter des commentaires associs qui seront visibles par les administrateurs, les superviseurs du workflow et les autres participants au workflow de donnes. Tant que vous tes toujours en train de travailler sur le bon de travail, vous pouvez diter ce commentaire.

Quand vous avez ralis toutes les actions demandes par le bon de travail, vous devez signaler la fin du travail en cliquant soit sur le bouton **Accepter**, soit sur le bouton **Rejeter**. Les libells de ces deux boutons peuvent varier en fonction du contexte du bon de travail

Pour suivre l'avancement du workflow de donnes associ au bon de travail qui vous est destin dans votre boîte de rception, cliquez sur le bouton "Afficher" dans la colonne 'Workflow de donnes' de la table. Une pop-up affichera une vue graphique interactive du workflow de donnes jusqu'au moment actuel ainsi que les tapes venir. Vous pouvez visualiser les dtails d'une tape en slectionnant l'tape.

> **Note**
>
> Si vous interrompez la session actuelle pendant un bon de travail, par exemple en fermant le navigateur ou en dconnectant, l'tat actuel du bon de travail est prserv. Quand vous revenez sur le bon de travail, il continue partir du mme point.

30.3 **Priorit de bons de travail**

Les bons de travail peuvent porter une priorit, qui peut tre utile pour trier et filtrer les bons de travail complter. La priorit d'un bon de travail est dfinie au niveau de son workflow de donnes, et n'est pas spcifique au bon de travail lui-mme. Par consquent, si le workflow de donnes est considr comme urgent, tous les bons de travail ouverts associs sont aussi considrs comme urgent. Par dfaut, il y a six niveaux de priorit, de "Trs peu prioritaire" "Urgent". Cependant, la reprsentation visuelle et le nommage des priorits dpendent de la configuration de votre rfrentiel TIBCO EBX.

Voir aussi*tche utilisateur (glossaire)* *[p 30]*

Concepts apparentés*Tche utilisateur* *[p 154]*

CHAPITRE **31**

Lancement et monitoring de workflows de donnes

Ce chapitre contient les sections suivantes :

1. Lancement d'un workflow de donnes
2. Activits de monitoring
3. Gestion de l'allocation des bons de travail

31.1 Lancement d'un workflow de donnes

Quand un modle de workflow vous autorise lancer des workflows de donnes depuis ses publications, vous pouvez crer de nouveaux workflows en utilisant la vue 'Lanceurs de workflow' dans le panneau de navigation. Pour crer un nouveau workflow de donnes, depuis une publication de modle de workflow, cliquer sur le bouton 'Lancer' de la ligne de la publication cible dans la table.

Vous pouvez alors dfinir des libells et descriptions localiss pour le nouveau workflow de donnes que vous lancez.

31.2 Activits de monitoring

Quand un modle de workflow vous donne les permissions de monitoring de workflow, vous avez la possibilit de suivre l'avancement des workflows de donnes qui sont en cours d'excution. Vous pouvez accder aux vues de monitoring, dans la section 'Monitoring' du panneau de navigation. Si vous disposez galement de permissions d’administration de workflow, vous pouvez effectuer les actions autorises associes depuis ces vues.

Lorsqu'un workflow de donnes, que vous tes autoris suivre, a fini son excution, il est affich dans 'Workflows termins', o vous pouvez consulter son historique.

31.3 Gestion de l'allocation des bons de travail

Lorsque vous tes autoris grer l'allocation des bons de travail, vous pouvez allouer manuellement des bons de travail durant l'excution des workflows associs au modle de workflow. Dans ce cas, vous pouvez effectuer une ou plusieurs des actions ci-dessous sur les bons de travail.

Slectionnez 'Bons de travail' dans la section 'Monitoring' du panneau de navigation. Les actions que vous pouvez effectuer sont affiches dans le menu 'Actions' de l'entre du bon de travail, selon son tat actuel, dans la table.

Allouer	Allouer un bon de travail un utilisateur spcifique. Cette action est disponible pour les bons de travail dans l'tat *propos*.
Dsallouer	Remettre un bon de travail qui est actuellement dans l'tat *allou* dans l'tat *propos*.
Rallouer	Modifier l'utilisateur qui le bon de travail est allou. Cette action est disponible pour les bons de travail dans l'tat *allou*.

Voir aussi

Bons de travail *[p 185]*

Permissions sur les workflows de donnes associs *[p 169]*

Concepts apparentés*Modles de workflow* *[p 148]*

CHAPITRE 32

Administration de workflows de donnes

Si vous disposez de permissions pour administrer des workflows de donnes, les vues 'Publications', 'Workflows actifs', et 'Bons de travail' associs seront accessibles sous le menu 'Monitoring' du panneau de navigation. Dans ces vues, sous les menus 'Actions' sur les lignes des tables, vous pourrez accder aux actions d'administration.

> **Note**
>
> Quand un modle de donnes vous donne des droits d'administration, vous aurez automatiquement les permissions de monitoring sur tous les objets associs l'excution de workflow, comme les publications, les workflows actifs, et les bons de travail.

Ce chapitre contient les sections suivantes :

1. Prsentation de l'excution de workflow de donnes
2. Actions d'administration de workflow de donnes

32.1 Prsentation de l'excution de workflow de donnes

Quand un workflow de donnes est lanc, un *jeton* qui reprsente l'tape en cours d'excution est cr et positionn au dbut du workflow. A chaque fois qu'une tape est termine, ce jeton se dplace sur la prochaine tape dfinie par le modle de workflow associ la publication du workflow de donnes.

Pendant l'excution d'un workflow de donnes, le jeton est positionn sur un des types d'tape suivants:

- une tche automatique, qui est lance automatiquement et n'a pas besoin d'interaction utilisateur. La tche automatique est termine quand les actions dfinies finissent leur excution.
- une tche utilisateur, qui gnre un ou plusieurs bons de travail effectus manuellement par les utilisateurs. Chaque bon de travail est termin pas une action 'Accepter' ou 'Rejeter', ralise explicitement par l'utilisateur. La fin de la tche utilisateur chapeau est dtermine en fonction du critre de fin de tche dfini pour la tche utilisateur dans le modle de workflow.
- une condition, qui est value automatiquement afin de dterminer l'tape suivante de l'excution du workflow de donnes.
- invocation de sous-workflows qui lance les sous-workflows associs et attend que les sous-workflows en cours soient termins.
- tche d'attente qui met en pause le workflow jusqu' ce qu'un vnement spcifique soit reu.

Le jeton peut tre dans les tats suivants :

- **A executer :** Le jeton est en train de passer la prochaine tape, en se basant sur le modle de workflow.
- **En cours d'excution :** Le jeton est positionn sur une tche automatique ou une condition en train de s'excuter.
- **Utilisateur :** Le jeton est positionn sur une tche utilisateur et attend une action utilisateur.
- **En attente de sous-workflows :** Le jeton est positionn sur une invocation de sous-workflows et attend la terminaison de tous les sous-workflows lancs.
- **En attente d'vnement :**Le jeton est positionn sur une tche d'attente et attend de recevoir un vnement donn.
- **Termin :** Le jeton a atteint la fin du workflow de donnes.
- **Erreur :** Une erreur est survenue.

Voir aussi*Workflow management* *[p 433]*

32.2 Actions d'administration de workflow de donnes

Actions sur les publications

Dsactivation d'une publication de workflow

Afin d'viter que de nouveaux workflows de donnes soient lancs depuis une publication de workflow, vous pouvez dsactiver la publication. Slectionnez la vue 'Publications' dans la panneau de navigation, puis slectionnez *Actions* > *Dsactiver* sur la ligne de la publication cible.

Une fois dsactive, la publication n'apparaîtra plus dans la vue 'Lanceurs de workflow' des utilisateurs. Toutefois, les workflows de donnes dj lancs vont continuer s'excuter.

> **Note**
>
> Suite la dsactivation d'une publication, il n'est pas possible de la ractiver partir de la section 'Workflows de donnes'. Seul un utilisateur avec le rle built-in 'Administrateur' peut ractiver une publication inactive dans la section 'Administration'. Cependant, il n'est pas conseill de modifier les tables techniques manuellement, car il est important de prserver l'intgrit des donnes techniques des workflows.

Dpublication d'une publication de workflow

Si une publication de workflow n'est plus utilise, vous pouvez la supprimer de toutes les vues de la section 'Workflows de donnes' en la dpubliant. Pour faire cela,

1. Dsactivez la publication de workflow afin d'viter que des utilisateurs continuent de lancer des nouveaux workflows de donnes sur cette publication. Pour cela, suivez le processus dcrit dans la section Dsactivation d'une publication de workflow [p 194].

2. Dpublier la publication de workflow en slectionnant *Actions* > *Dpublier* de la ligne de la publication cible.

> **Note**
>
> A la dpublication d'une publication de workflow, une confirmation vous sera demande pour terminer et purger tous les workflows de donnes en cours qui ont t lancs depuis cette publication de workflow, ainsi que tout bon de travail associ. Toute perte de donnes rsultant d'une fin prmature est alors dfinitive.

Actions sur workflows de donnes

Dans les vues tabulaires des workflows de donnes, chaque enregistrement porte un menu *Actions* qui permet d'excuter des services sur un workflow de donnes.

R-excution d'une tape

Dans le cas d'une erreur inattendue pendant l'excution d'une tape, par exemple, cause d'un problme de permissions ou de ressources non disponibles, vous pouvez "rejouer" une tape en tant qu'administrateur de workflow. En rejouant une tape, l'environnement d'excution associ est nettoy, notamment les bons de travail et sous-workflows lis, et le jeton est repositonn au debut de l'tape courante.

Pour rejouer l'tape courante dans un workflow de donnes, slectionnez *Actions* > *Rejouer l'tape* dans la ligne du workflow cible dans la table 'Workflows actifs'.

Terminer un workflow de donnes actif et le purger

Pour terminer un workflow de donnes en cours d'excution, slectionnez *Actions* > *Terminer et purger* dans la ligne du workflow cible dans la table 'Workflows actifs'. L'action stoppe l'excution du workflow de donnes et supprime le workflow, tous les bons de travail et sous-workflows associs.

> **Note**
>
> Cette action n'est pas disponible pour les workflows dans l'tat 'En cours d'excution' et pour les sous-workflows lancs par d'autres workflows.

> **Note**
>
> Les historiques du workflow ne sont pas supprims.

Forcer la terminaison d'un workflow de donnes actif

Pour forcer la terminaison d'un workflow de donnes en cours d'excution, slectionnez *Actions* >*Forcer la terminaison* dans la ligne du workflow cible dans la table 'Workflows actifs'. L'action stoppe l'excution du workflow de donnes et supprime les ventuels bons de travail et sous-workflows associs.

> **Note**
>
> Cette action est disponible pour les sous-workflows, et pour les workflows en erreur bloqus sur la dernire tape.

> **Note**
>
> Les historiques du workflow ne sont pas supprims.

Forcer le rveil d'un workflow en attente

Pour rveiller un workflow qui est en attente d'vnement, slectionner **Actions > Forcer le rveil** partir du workflow dans la table 'Workflows actifs'. Cela entraîne le rveil du workflow. Avant d'effectuer cette action, l'administrateur doit mettre jour le contexte de donnes afin de s'assurer que le workflow peut excuter les tches suivantes.

> **Note**
>
> Cette action est disponible uniquement pour les workflows qui sont l'tat 'en attente d'vnement'.

Purge d'un workflow de donnes termin

Quand un workflow de donnes a termin son excution, son historique est visible pour ses superviseurs et administrateurs dans la vue 'Workflows termins'. Pour purger le workflow termin, vous pouvez effectuer un nettoyage en slectionnant *Actions > Purger* dans la ligne du workflow cible de la table 'Workflows termins'.

Un workflow purg n’est plus visible dans la vue 'Workflows termins'. Cependant, son historique reste consultable dans la zone d’administration technique.

> **Note**
>
> Cette action n'est pas disponible pour les sous-workflows lancs par d'autres workflows.

Voir aussi*Workflow management* *[p 433]*

Modification de la priorit d'un workflow de donnes

Suite au lancement d'un workflow de donnes, un administrateur du workflow peut modifier son niveau de priorit. En modifiant la priorit du workflow de donnes, la priorit de tous les bons de travail existants et venir de ce workflow sera modifie. Pour modifier la priorit d'un workflow de donnes, slectionnez *Actions > Modifier la priorit* dans la ligne du workflow cible dans la table 'Workflows actifs'.

Voir aussi*Permissions sur les workflows de donnes associs* *[p 169]*

Services de données

CHAPITRE **33**

Introduction aux services de donnes

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Utilisation de l'interface utilisateur de la section Services de donnes

33.1 Prsentation

Fonction du service de donnes

Un service de donnes [p 31] est :

- un web service standard qui permet d'interagir avec TIBCO EBX.

  Les services de donnes SOAP peuvent tre gnrs dynamiquement partir d'un modle de donnes dans la section 'Services de donnes'.
- un service REST qui permet d'interroger le contenu du rfrentiel EBX.

  Un service RESTful prdfini ne ncessite pas d'interface de service, il est auto-descriptif par l'intermdiaire des mta-donnes retournes.

Ils peuvent tre utiliss pour accder une partie des fonctionnalits disponibles par l'interface utilisateur.

Voir aussi

WSDL/SOAP [p 620]

REST [p 674]

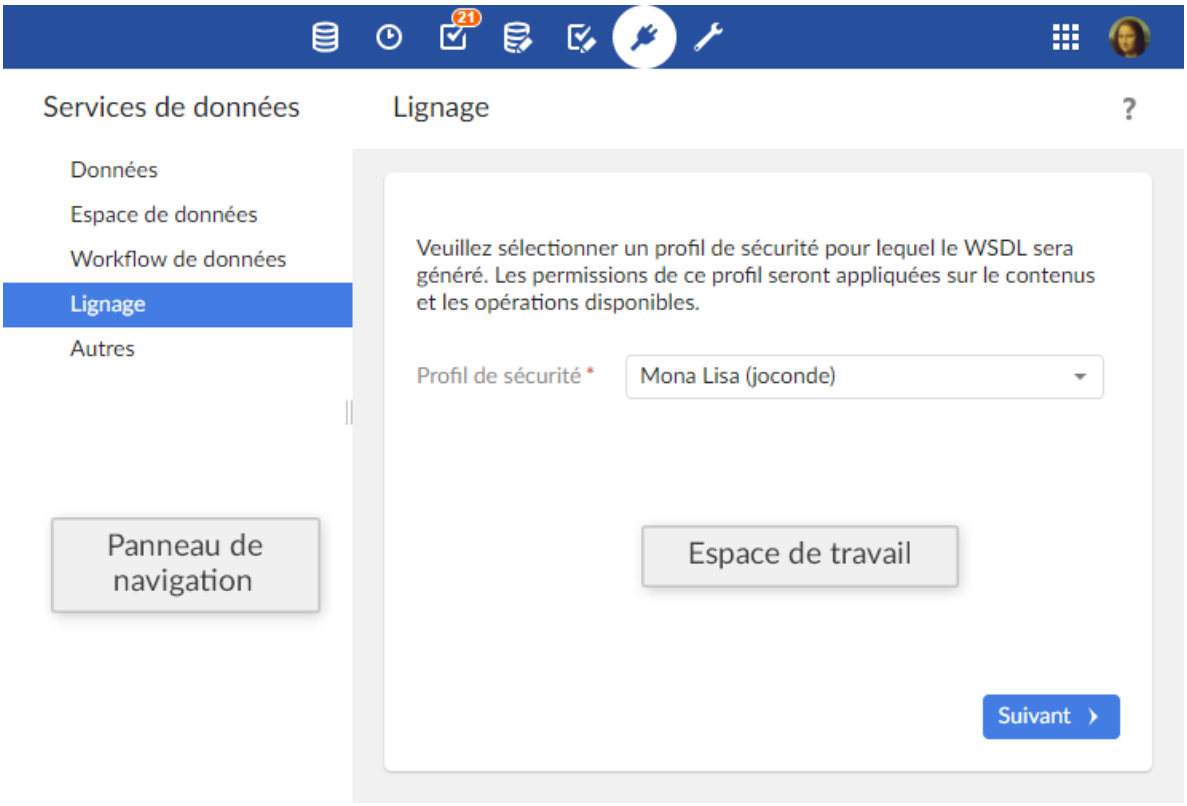
Lignage

Le lignage [p 32] tablit des profils de droit d'accs utiliss par les services de donnes. Quand les services de donnes accdent aux interfaces WSDL, ils utilisent les profils de droit d'accs dfinis par ce mcanisme.

Glossaire

Voir aussi*Services de donnes [p 31]*

33.2 Utilisation de l'interface utilisateur de la section Services de donnes



> **Note**
>
> Seuls les utilisateurs autoriss peuvent accder cette section via la 'Perspective avance'.

Concepts apparentés

Espaces de donnes *[p 98]*

Jeux de donnes *[p 120]*

Workflows de donnes *[p 176]*

Introduction to data services *[p 620]*

CHAPITRE 34

Gnration de WSDL pour services de donnes

Ce chapitre contient les sections suivantes :

1. Gnrer un WSDL pour accder aux donnes
2. Gnrer un WSDL pour accder un espace de donnes
3. Gnrer un WSDL pour contrler un workflow de donnes
4. Gnrer un WSDL pour un lignage
5. Gnrer un WSDL pour l'administration
6. Gnrer un WSDL pour modifier l'annuaire par dfaut

34.1 Gnrer un WSDL pour accder aux donnes

La gnration de WSDL pour l'accs aux donnes est disponible en slectionnant 'Donnes' dans la section 'Services de donnes'.

Les tapes de la gnration d'un WSDL sont les suivantes :

1. Slectionner si le WSDL sera utilis pour des oprations sur un jeu de donnes ou sur une table.
2. Identifier l'espace de donnes et le jeu de donnes cibls par les oprations.
3. Slectionner les tables sur lesquelles les oprations sont autorises, ainsi que les oprations permises.
4. Tlcharger le fichier WSDL gnr en cliquant sur le bouton 'Tlcharger le WSDL'.

Oprations disponibles sur un jeu de donnes

Les oprations suivantes sur les jeux de donnes sont disponibles en utilisant le WSDL gnr :

- Slectionner les donnes d'un jeu de donnes pour un espace de donnes ou une image.
- Rcuprer les changements du jeu de donnes entre espaces de donnes ou images
- Actualiser une unit de rplication en base de donnes

Voir aussi

WSDL download from HTTP protocol *[p 633]*

Operations generated from a data model *[p 639]*

Oprations disponibles sur une table d'un jeu de donnes

Si slectionnes, les oprations suivantes sur les tables sont disponibles en utilisant le WSDL gnr :

- Slectionner un (ou plusieurs) enregistrement(s)
- Insrer un (ou plusieurs) enregistrement(s)
- Mettre jour un (ou plusieurs) enregistrement(s)
- Supprimer un (ou plusieurs) enregistrement(s)
- Compter des enregistrements
- Rcuprer les changements entre espaces de donnes ou images
- Obtenir les droits d'accs

Voir aussi

WSDL download from HTTP protocol [p 633]

Operations generated from a data model [p 639]

34.2 Gnrer un WSDL pour accder un espace de donnes

La gnration de WSDL pour la manipulation d'un espace de donnes est accessible en slectionnant 'Espace de donnes' dans la section 'Services de donnes'. Le WSDL gnr n'est pas spcifique un espace de donnes et aucune information n'est requise. Il peut tre tlcharg grce au bouton **Tlcharger le WSDL**.

Oprations disponibles sur un espace de donnes

Les oprations suivantes sur les espaces de donnes sont disponibles en utilisant le WSDL gnr :

- Crer un espace de donnes
- Crer une image
- Fermer un espace de donnes
- Fermer une image
- Fusionner un espace de donnes
- Valider un espace de donnes ou une image
- Valider un jeu de donnes
- Verrouiller un espace de donnes
- Dverrouiller un espace de donnes

Voir aussi

WSDL download from HTTP protocol [p 633]

Operations on datasets and dataspaces [p 660]

34.3 Gnrer un WSDL pour contrler un workflow de donnes

La gnration d'un WSDL pour le contrle d'un workflow est accessible en slectionnant 'Workflow de donnes' dans la section 'Services de donnes'. Le WSDL gnr n'est pas spcifique une publication de workflow et aucune information n'est requise. Il peut tre tlcharg grce au bouton **Tlcharger le WSDL**.

Oprations disponibles pour contrler un workflow de donnes

Les oprations suivantes sur les espaces de donnes sont disponibles en utilisant le WSDL gnr :

- Dmarrer un workflow
- Rveiller un workflow
- Terminer un workflow

Voir aussi

WSDL download from HTTP protocol [p 633]

Operations on data workflows [p 666]

34.4 Gnrer un WSDL pour un lignage

La gnration d'un WSDL pour un lignage est accessible en slectionnant 'Lignage' dans la section 'Services de donnes', sous rserve que des profils aient t autoriss par un profil administrateur dans *Administration > Lignage*.

Les WSDL gnrs pour accder aux tables sont les mmes que ceux utiliss pour la gnration d'un WSDL d'accs aux donnes [p 201].

Les tapes de la gnration de ce WSDL sont les suivantes :

1. Slectionner un rle ou un utilisateur, dont les permissions seront appliques. Un rle ou un utilisateur doit tre autoris tre utilis pour le lignage par un administrateur.
2. Identifier l'espace de donnes et le jeu de donnes cibls par les oprations.
3. Slectionner les tables sur lesquelles les oprations sont autorises, ainsi que les oprations permises.
4. Tlcharger le fichier WSDL gnr en cliquant sur le bouton **Tlcharger le WSDL**.

Voir aussi*Lignage [p 198]*

34.5 Gnrer un WSDL pour l'administration

Cette action ne peut tre effectue que par un administrateur.

La gnration d'un WSDL pour :

- modifier l'accs l'interface utilisateur
- rcuprer les informations systme

est disponible en slectionnant 'Administration' dans la section 'Services de donnes'.

Oprations disponibles pour l'administration

- Fermer l'interface utilisateur

- Ouvrir l'interface utilisateur
- Obtenir les informations systme

Voir aussi

WSDL download from HTTP protocol *[p 633]*

User interface operations *[p 670]*

System information operation *[p 670]*

34.6 Gnrer un WSDL pour modifier l'annuaire par dfaut

Cette action ne peut tre effectue que par un administrateur et seulement si l'annuaire par dfaut est utilis.

La gnration d'un WSDL pour modifier l'annuaire d'accs est accessible en slectionnant 'Annuaire' dans la section 'Services de donnes'.

Oprations disponibles pour modifier l'annuaire par dfaut

Les WSDL gnrs pour accder aux tables sont les mmes que ceux utiliss pour la gnration d'un WSDL d'accs aux donnes [p 201].

Voir aussi

WSDL download from HTTP protocol *[p 633]*

Directory services *[p 669]*

Manuel de référence

Intégration

CHAPITRE 35

Prsentation de l'intgration et des dploiements

Il existe diffrents services et composants API permettant de dvelopper des extensions personnalises pour TIBCO EBX et de les intgrer d'autres systmes.

Ce chapitre contient les sections suivantes :

1. Utiliser EBX comme composant web
2. Personnalisation de l'interface utilisateur
3. Services de donnes
4. Services d'import/export XML et CSV
5. Services programmatiques

35.1 Utiliser EBX comme composant web

Il est possible d'utiliser EBX comme composant web d'une interface utilisateur, en lanant une requte HTTP. Ces composants web peuvent tre intgrs n'importe quelle application accessible via un navigateur web compatible [p 334].

Les crans EBX sont gnralement intgrs la structure intranet de l'organisation. Les composants web peuvent aussi tre lancs partir de l'interface utilisateur d'EBX dans les Services utilisateurs [p 207].

Voir aussi*Utiliser EBX comme composant web [p 211]*

35.2 Personnalisation de l'interface utilisateur

Services utilisateur (user services)

Un service utilisateur est une extension d'EBX qui fournit une interface homme-machine (IHM) permettant aux utilisateurs d'accder des fonctionnalits spcifiques ou avances.

Voir aussi*Vue d'ensemble des services utilisateurs [p 587]*

Mise en page personnalise

Une couche de prsentation permet de surcharger la mise en page par dfaut des formulaires dans l'interface utilisateur.

Voir aussi*Form layout* *[p 585]*

Widgets personnaliss

Un widget personnalis (custom widget) est un composant graphique dvelopp spcifiquement pour personnaliser l'affichage de groupes ou de champs d’un modle de donnes ou d'un schma programmatique.

Voir aussi*Custom widgets* *[p 585]*

Ajax

EBX supporte Ajax pour les changes de donnes asynchrones avec le serveur sans avoir rafraîchir la page courante.

Voir aussi

User service Ajax callbacks *[p 584]*

Ajax component `UIAjaxComponent`[API]

Spcifier les filtres UI sur une table

En complment des filtres par dfaut et des panneaux de recherche de l'interface utilisateur, il est possible de dfinir des filtres additionnels en fonction de la structure de la table. Pour cela, cette classe spcifique doit tre prcise lors de la dfinition de la table et doit tendre `UITableFilter`.

Voir `UITableFilter`[API] pour plus d'informations.

35.3 Services de donnes

Le module 'Services de donnes' permet aux systmes externes d'interagir avec EBX en utilisant, l'une des interfaces suivantes:

- Web Services Description Language (WSDL)
- Representational state transfer (REST)

Voir aussi

WSDL/SOAP *[p 620]*

REST *[p 674]*

35.4 Services d'import/export XML et CSV

EBX contient des services intgrs pour l'import et l'export de donnes aux formats XML et CSV. Les imports et les exports en XML et CSV peuvent tre effectus via l'interface utilisateur, les services de donnes, ou l'API Java.

Voir aussi

Import et export XML *[p 235]*

Import et export CSV *[p 241]*

35.5 **Services programmatiques**

Les services programmatiques permettent d'excuter des procdures dans un contexte prcis, par exemple dans une tche planifie ou un traitement par lot.

Quelques exemples de services programmatiques :

- Import de donnes partir d'une source externe,
- Export de donnes vers des systmes multiples,
- Historisation des donnes, lance par un systme de supervision,
- Optimisation et remaniement des donnes si les **services intgrs d'optimisation** `AdaptationTreeOptimizerSpec`[API] d'EBX ne suffisent pas.

Voir aussi *`ProgrammaticService`*[API]

CHAPITRE **36**

Utiliser TIBCO EBX comme composant web

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Intgration des composants web d'EBX des applications
3. Slection de l'lment de rfrentiel et du primtre
4. Slection combine
5. Caractristiques de la requte
6. Exemples d'appels un composant web EBX

36.1 Prsentation

EBX peut tre utilis comme composant web d'une interface utilisateur, appel par protocole HTTP. Un composant web EBX peut tre intgr n'importe quelle application accessible via un navigateur web compatible. Ce mode d'accs permet de profiter des fonctionnalits majeures d'EBX, telles que l'authentification utilisateur, la validation des donnes, ainsi que la gnration automatique de l'interface utilisateur, tout en permettant galement d'axer la navigation utilisateur sur certains lments du rfrentiel.

Les composants web d'EBX sont gnralement intgrs la structure intranet de l'organisation ou aux applications qui grent l'attribution de tches spcifiques aux utilisateurs.

Voir aussi*Navigateurs web compatibles* *[p 334]*

36.2 Intgration des composants web d'EBX des applications

Une application web qui appelle un composant web d'EBX peut tre :

1. Une application non-Java, le cas le plus basique tant une page HTML statique.

   Dans ce cas, l'application doit envoyer une requte HTTP qui respecte les spcifications de requte [p 213] du composant web EBX.
2. Une application Java, par exemple :

- Une application web Java excute sur la mme instance de serveur d'application que le rfrentiel EBX qu'il rfrence ou sur une instance de serveur d'application diffrente.
- Un Service utilisateur [p 207] ou un Widget personnalis [p 208] EBX, auquel cas la nouvelle session hritera automatiquement de la session parent d'EBX.

> **Note**
>
> En Java, la mthode recommande pour tablir des requtes HTTP qui appellent les composants web d'EBX est la classe `UIHttpManagerComponent`[API] dans l'API.

36.3 Slection de l'lment de rfrentiel et du primtre

Quand un composant web d'EBX est appel, l'utilisateur doit d'abord tre authentifi dans la session HTTP nouvellement instancie. Le composant web slectionne alors un lment du rfrentiel et l'affiche conformment au paramtre de mise en page du `scope` dfini dans la requte.

Le paramtre `firstCallDisplay` peut changer cet affichage automatique en fonction de sa valeur.

Les lments du rfrentiel qui peuvent tre slectionns sont les suivants :

- Espace de donnes ou image
- Jeu de donnes
- Noeud
- Table ou vue publie
- Enregistrement de table

Le primtre, dfini par le paramtre 'scope', dtermine quelle partie de l'interface est affiche l'utilisateur et dfinit les limites de navigation de l'utilisateur pendant la session. Le primtre par dfaut utilis par le composant web est le plus petit possible dpendant de l'entit ou du service slectionns ou invoqus par la requte.

Voir aussi*scope* *[p 217]*

Voir aussi*firstCallDisplay* *[p 217]*

Il est galement possible de slectionner une perspective particulire ainsi qu’une action de perspective.

Par dfaut, la slection de l'lment est faite dans le cadre de la perspective de l'utilisateur si le primtre de l'affichage ("scope") est complet ("full").

Voir aussi*Perspective* *[p 17]*

36.4 Slection combine

Une URL de composant Web peut spcifier une perspective ainsi qu’une action ou une entit (espace de donnes, jeu de donnes, etc). Ainsi, pour un composant Web ayant dans son URL une perspective et une entit (mais pas d’action), si une action de la perspective correspond cette entit, alors cette action est automatiquement slectionne.

Sinon, si aucune action ne correspond cette entit, aucune action n'est slectionne mais l’entit est ouverte.

Si une action est spcifi en mme temps qu’une entit, cette dernire est ignore et l’action sera slectionne.

Cas particulier

Si l'entit cible est un enregistrement et qu'une action correspond la table de l'enregistrement, alors cette action est slectionne et l'enregistrement sera ouvert l'intrieur de l'action.

De mme, si un bon de travail est cibl par le composant Web, et si une action de type « inbox » existe dans la perspective, alors cette action sera slectionne et le bon de travail sera ouvert l'intrieur de l'action.

Limitations connues

Si le composant Web spcifie un prdicat pour filtrer une table, il faut que l'action de perspective spcifie exactement le mme prdicat pour pouvoir tre slectionne.

De mme, si l'action spcifie un prdicat de filtrage sur une table, le composant Web doit spcifier exactement le mme prdicat pour que la correspondance soit tablie.

36.5 **Caractristiques de la requte**

URL de base

Dans un dploiement par dfaut, l'URL de base doit avoir la forme suivante :

```
http://<host>[:<port>]/ebx/
```

> **Note**
>
> L'URL de base doit faire rfrence au servlet `FrontServlet`, dfini dans le descripteur de dploiement `/WEB-INF/web.xml` de l'application web `ebx.war`.

Authentification de l'utilisateur et paramtres d'informations de la session

Paramtre	Description	Requis
`login` et `password`, ou un *jeton utilisateur li au rpertoire*	Dfinit les proprits d'authentification de l'utilisateur. Si le login/mot de passe ou le jeton utilisateur li au rpertoire ne sont pas fournis, l'utilisateur sera invit s'authentifier via la page de connexion du rfrentiel. Voir `Directory`[API] pour plus d'informations.	Non
`trackingInfo`	Dfinit les informations de suivi de la nouvelle session. Les informations de suivi sont recueillies dans les tables de l'historique. Paralllement, ces informations peuvent tre utilises pour restreindre programmatiquement les autorisations d'accs. Voir `AccessRule`[API] pour plus d'informations.	Non
`redirect`	L'URL vers laquelle l'utilisateur sera redirig la fin de la session du composant, lorsqu'il clique sur le bouton 'Fermer'. Le bouton 'Fermer' est systmatiquement affich lors de la slection d'enregistrements, cependant il faut toujours spcifier via le paramtre `closeButton` si l'affichage doit tre maintenu dans les autres cas. Pour plus d'informations, voir Exit policy [p 411].	Non
`locale`	Spcifie la locale utiliser. La valeur est soit `en-US` soit `fr-FR`.	Non, la locale par dfaut est celle enregistre pour l'utilisateur.

Entit et paramtres de slection du service

Paramtre	Description	Requis
`branch`	Slectionne l'espace de donnes spcifi.	Non
`version`	Slectionne l'image d'espace de donne spcifie.	Non
`instance`	Slectionne le jeu de donnes spcifi. La valeur doit tre la rfrence d'un jeu de donnes qui existe dans l'espace de donnes ou dans l'image slectionns.	Seulement si `xpath` ou `viewPublication` sont spcifis.
`viewPublication`	Dfinit le nom de publication de la vue tabulaire ou hirarchique appliquer au contenu choisi. Ce nom de publication est celui dclar lors de la publication de la vue. Il se trouve dans l'espace 'Administration' sous *Configuration des vues > Vues*. Tous les paramtres de la vue, tels que ses filtres, l'ordre de tri et les colonnes affiches, sont appliqus au rsultat. Un espace de donnes et un jeu de donnes doivent tre slectionns afin que cette vue s'applique. Le choix d'une table cible n'est pas ncessaire, puisqu'elle peut tre automatiquement dtermine sur la base de la dfinition de la vue. Ce paramtre peut tre associ au prdicat spcifi dans le paramtre `xpath` en tant qu'opration logique 'AND'.	Non
`xpath`	Dfinit la slection d'un noeud dans le jeu de donnes. La valeur peut tre un chemin absolu valide situ dans le jeu de donnes slectionn. L'criture doit tre conforme un XPath simplifi, avec une syntaxe abrge. Il peut aussi s'agir d'un prdicat entour par "[" and "]" si une table peut tre automatiquement slectionne l'aide d'autres paramtres de composants web (par exemple, `viewPublication` ou `workflowView`). Voir aussi XPath supported syntax [p 249] pour la syntaxe XPath Voir `UIHttpManagerComponent.setPredicate`[API] pour plus d'informations.	Non
`service`	Dfinit le service auquel accder. Pour plus d'informations sur les services utilisateurs intgrs, voir Services built-in [p 221]. Dans l'API Java, voir `ServiceKey`[API] pour plus d'informations.	Non
`workflowView`	Dfinit la section du workflow slectionner. Voir `WorkflowView`[API] pour plus d'informations.	Non
`perspectiveName`	Dfinit le nom de la perspective slectionner. Si ce paramtre est dfini, le paramtre `scope` ne pourra prendre que deux valeurs:full et data.	Seulement si `perspectiveActionId` ou `perspectiveActionName` est spcifi.
`perspectiveActionId`	**Dprci.** Merci de considrer l'utilisation de `perspectiveActionName` la place. Dfinit l'identifiant de l'action de perspective slectionner.	Non

Paramtre	Description	Requis
`perspectiveActionName`	Dfinit le nom unique de l'action de perspective slectionner.	Non

Paramtres de mise en page

Paramtre	Description	Requis
`scope`	Dfinit le primtre utilis par le composant web. Il peut prendre la valeur `full`, `data`, `dataspace`, `dataset` ou `node`. Voir `UIHttpManagerComponent.Scope`[API] pour plus d'informations.	Non, la valeur par dfaut sera calcule pour tre la plus petite possible en fonction de la slection cible.

`firstCallDisplay`	Dfinit l'affichage qui doit tre utilis la place de celui dtermin par la combinaison de la slection et du paramtre `scope`.
	Les valeurs possibles sont :
	• `auto`:L'affichage est automatiquement positionn en fonction de la slection.
	• `view`:Force l'affichage de la vue tabulaire ou de la vue hirarchique.
	• `record`:Si le prdicat cible au moins un enregistrement, force l'affichage du premier enregistrement de la liste.
	Par exemple,
	`firstCallDisplay=view`
	`firstCallDisplay=view:hierarchyExpanded`
	`firstCallDisplay=record`
	`firstCallDisplay=record:{predicate}`
	Voir `UIHttpManagerComponent.setFirstCallDisplay`[API] pour plus d'informations.
	Voir `UIHttpManagerComponent.setFirstCallDisplayHierarchyExpanded`[API] pour plus d'informations.
	Voir `UIHttpManagerComponent.setFirstCallDisplayRecord`[API] pour plus d'informations.
		Non, la valeur par dfaut sera calcule en fonction de la slection cible.

`closeButton`	Dfinit la manire dont le bouton de fin de session sera affich. La valeur peut tre `logout` ou `cross`. Voir `UIHttpManagerComponent.CloseButtonSpec`[API] pour plus d'informations.	Non. Si le primtre n'est pas `full`, aucun bouton de fin de session ne sera affich par dfaut.

`dataSetFeatures`	Dfinit les fonctionnalits afficher dans un service utilisateur au niveau du jeu de donnes ou d'un formulaire en dehors d'une table.
	Ces options se rapportent uniquement aux fonctionnalits au sein de l'espace de travail. Il est recommand d'utiliser cette proprit avec le plus petit `scope` possible, c'est dire `dataset` ou `node`.
	Syntaxe :
	`<prefixe> ":" <feature> [ "," <feature>]*`
	o
	• `<prefixe>` est `hide` ou `show`,
	• `<feature>` est `services`, `title`, `save`, ou `revert`.
	Par exemple,
	`hide:title`
	`show:save,revert`
	Voir `UIHttpManagerComponent.DataSetFeatures`[API] pour plus d'informations.
		Non.

Paramtre	Description	Requis

`viewFeatures`	Dfinit les fonctionnalits afficher dans une vue tabulaire ou hirarchique (au niveau de la table).
	Ces options se rapportent uniquement aux fonctionnalits au sein de l'espace de travail. Il est recommand d'utiliser cette proprit avec le plus petit `scope` possible, c'est dire `dataset` ou `node`.
	Syntaxe :
	`<prefixe> ":" <feature> [ "," <feature>]*`
	o
	• `<prefixe>` est `hide` ou `show`,
	• `<feature>` est `create`, `views`, `selection`, `filters`, `services`, `refresh`, `title`, ou `breadcrumb`.
	Par exemple,
	`hide:title,selection`
	`show:service,title,breadcrumb`
	Voir `UIHttpManagerComponent.ViewFeatures`[API] pour plus d'informations.
		Non.

`recordFeatures`	Dfinit les fonctionnalits afficher dans un formulaire au niveau de l'enregistrement.
	Ces options se rapportent uniquement aux fonctionnalits au sein de l'espace de travail. Il est recommand d'utiliser cette proprit avec le plus petit `scope` possible, c'est dire `dataset` ou `node`.
	Syntaxe:
	`<prefixe> ":" <feature> [ "," <feature>]*`
	o
	• `<prefixe>` est `hide` ou `show`,
	• `<feature>` est `services`, `title`, `breadcrumb`, `save`, `saveAndClose`, `close`, ou `revert`.
	Par exemple,
	`hide:title`
	`show:save,saveAndClose,revert`
	Voir `UIHttpManagerComponent.RecordFeatures`[API] pour plus d'informations.
		Non.

`pageSize`	Prcise le nombre d'enregistrements affichs par page dans une vue table (qu'elle soit tabulaire ou hirarchique).	Non.
`startWorkItem`	Prcise qu'un lment de travail doit tre automatiquement pris et dmarr. La valeur peut tre `true` ou `false`. Voir `ServiceKey.WORKFLOW`[API] pour plus d'informations.	Non. La valeur par dfaut est `false`, l'tat du bon de travail cible restant inchang.

36.6 **Exemples d'appels un composant web EBX**

URI minimale :

`http://localhost:8080/ebx/`

Se connecte en tant qu"admin' et slectionne l'espace de donnes 'Rfrence' :

`http://localhost:8080/ebx/?login=admin&password=admin&branch=Reference`

Slectionne l'espace de donnes 'Rfrence' et accde au service de validation intgr :

```
http://localhost:8080/ebx/?
login=admin&password=admin&branch=Reference&service=@validation
```

Slectionne la table des rles dans le rpertoire par dfaut :

```
http://localhost:8080/ebx/?login=admin&password=admin&branch=ebx-
directory&instance=ebx-directory&xpath=/directory/roles
```

Slectionne l'enregistrement 'admin' dans le rpertoire par dfaut :

```
http://localhost:8080/ebx/?login=admin&password=admin&branch=ebx-
directory&instance=ebx-directory&xpath=/directory/users[./login="admin"]
```

Accde l'interface pour crer un nouvel utilisateur dans le rpertoire par dfaut :

```
http://localhost:8080/ebx/?login=admin&password=admin&branch=ebx-
directory&instance=ebx-directory&xpath=/directory/users&service=@creation
```

Compare l'enregistrement 'admin' du rpertoire par dfaut avec l'enregistrement 'jSmith' :

Compare l'enregistrement 'R1' du jeu de donnes 'instanceId' dans l'espace de donnes 'Reference' avec l'enregistrement 'R0' :

```
http://localhost:8080/ebx/?login=admin&password=admin&branch=ebx-
directory&instance=ebx-directory&xpath=/directory/users[./
login="admin"]&service=@compare&compare.branch=ebx-directory&compare.instance=ebx-
directory&compare.xpath=/directory/users[./login="jSmith"]
```

CHAPITRE 37

Services UI prdfinis

EBX inclut des services UI prdfinis. Ces services peuvent tre utiliss :

- pour dfinir une tche utilisateur [p 154]
- pour dfinir une action dans le menu d'une perspective [p 18]
- comme base pour dfinir un service UI tendu [p 611]
- pour utiliser EBX en mode composant Web [p 211]

Cette page rfrence tous les services UI prdfinis et leurs paramtres.

Ce chapitre contient les sections suivantes :

1. Accder des donnes (service par dfaut)
2. Crer un nouvel enregistrement
3. Dupliquer un enregistrement
4. Exporter les donnes depuis une table au format XML
5. Exporter les donnes depuis une table au format CSV
6. Importer les donnes dans une table depuis un fichier XML
7. Importer les donnes dans une table depuis un fichier CSV
8. Accder un espace de donnes
9. Valider un espace de donnes, une image ou un jeu de donnes
10. Fusionner un espace de donnes
11. Accder l'interface de fusion des espace de donnes
12. Comparer deux contenus
13. Workflows de donnes

37.1 Accder des donnes (service par dfaut)

Par dfaut, un workflow considre automatiquement ce service comme termin. Le bouton "Accepter" est donc toujours disponible.

Ce service est utilis par dfaut si aucun service n'est spcifi.

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`disableAutoComplete`	Dsactiver Accepter au dmarrage	Par dfaut, l'interaction associe ce service est directement considre comme termine. Par consquent, le bouton Accepter est automatiquement affich l'ouverture du bon de travail. Ce paramtre est utile pour dsactiver ce comportement. Si la valeur est "true", le dveloppeur a la charge de terminer l'interaction en utilisant SessionInteraction dans un service UI ou un trigger, par exemple. La valeur par dfaut est "false". Les perspectives n'utilisent pas ce paramtre.
`firstCallDisplay`	Mode d'affichage du premier appel	Dfinit le mode d'affichage qui doit tre utilis lorsqu'une table filtre ou un enregistrement est affich lors du premier appel. Dfaut (valeur = 'auto'):l'affichage est dfini automatiquement en fonction de la slection. Vue (valeur = 'view'):force l'affichage de la vue tabulaire ou hirarchique. Enregistrement (valeur = 'record'):si le prdicat a au moins un enregistrement, force l'affichage du formulaire de l'enregistrement.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes est obligatoire pour ce service.
`viewPublication`	Vue	Le nom de publication de la vue afficher. La vue doit tre configure pour la table slectionne.

Paramtre	Libell	Description
`xpath`	Noeud du jeu de donnes (XPath)	La valeur saisie doit tre un chemin absolu d'un noeud dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge.

37.2 Crer un nouvel enregistrement

Pour un workflow, le service creation est considr termin aprs la premire soumission sans erreur (enregistrement cr). Si ce service est appel alors qu'il est dj termin, l'enregistrement cr est affich en mode modification ou lecture seule (selon les droits de l'utilisateur).

Paramtre d'appel du service : `service=@creation`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Ce champ est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ est obligatoire pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`xpath`	Table du jeu de donnes (XPath)	La valeur saisie doit tre un chemin absolu d'une table dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

Paramtres en sortie

Paramtre	Libell	Description
`created`	Enregistrement cr	Contient le xPath de l'enregistrement cr.

37.3 Dupliquer un enregistrement

Pour un workflow, le service duplicate est considr termin aprs la premire soumission sans erreur (enregistrement cr). Si ce service est appel alors qu'il est dj termin, l'enregistrement cr est affich en mode modification ou lecture seule (selon les droits de l'utilisateur).

Paramtre d'appel du service : `service=@duplicate`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Ce champ est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ est obligatoire pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`xpath`	Enregistrement dupliquer (XPath)	La valeur saisie doit tre un chemin absolu d'un enregistrement existant. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

Paramtres en sortie

Paramtre	Libell	Description
`created`	Enregistrement cr	Contient le xPath de l'enregistrement cr.

37.4 Exporter les donnes depuis une table au format XML

Le service exportToXML est considr termin une fois l'export termin et le fichier tlcharg.

Paramtre d'appel du service : `service=@exportToXML`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ doit tre renseign pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes est obligatoire pour ce service.
`xpath`	Table du jeu de donnes exporter (XPath)	La valeur saisie doit tre un chemin absolu d'une table dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

37.5 **Exporter les donnes depuis une table au format CSV**

Un workflow considre automatiquement le service exportToCSV comme termin une fois l'export achev et le fichier tlcharg.

Paramtre d'appel du service : `service=@exportToCSV`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ doit tre renseign pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes est obligatoire pour ce service.
`xpath`	Table du jeu de donnes exporter (XPath)	La valeur saisie doit tre un chemin absolu d'une table dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

37.6 **Importer les donnes dans une table depuis un fichier XML**

Un workflow considre le service importFromXML comme termin une fois l'import russi.

Paramtre d'appel du service : `service=@importFromXML`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ doit tre renseign pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`xpath`	Table du jeu de donnes dans laquelle importer (XPath)	La valeur saisie doit tre un chemin absolu d'une table dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

37.7 **Importer les donnes dans une table depuis un fichier CSV**

Un workflow considre le service importFromCSV comme termin une fois l'import russi.

Paramtre d'appel du service : `service=@importFromCSV`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi - Ce champ doit tre renseign pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`xpath`	Table du jeu de donnes dans laquelle importer (XPath)	La valeur saisie doit tre un chemin absolu d'une table dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge - Ce champ est obligatoire pour ce service.

37.8 **Accder un espace de donnes**

Un workflow considre automatiquement le service de slection d'espace de donnes comme termin.

Paramtre d'appel du service : `service=@selectDataSpace`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes est obligatoire pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes est obligatoire pour ce service.

37.9 **Valider un espace de donnes, une image ou un jeu de donnes**

Un workflow considre automatiquement le service validation comme termin.

Paramtre d'appel du service : `service=@validation`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes ou une image est obligatoire pour ce service.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes ou une image est obligatoire pour ce service.

Paramtres en sortie

Paramtre	Libell	Description
`hasError`	Erreurs trouves	Contient 'true' si la validation a gnr des erreurs.
`hasFatal`	Erreurs fatales trouves	Contient 'true' si la validation a gnr des erreurs fatales.
`hasInfo`	Informations trouves	Contient 'true' si la validation a gnr des informations.
`hasWarning`	Avertissements trouves	Contient 'true' si la validation a gnr des avertissements.

37.10 **Fusionner un espace de donnes**

Un workflow considre le service merge comme termin une fois la fusion effectue et l'espace de donnes ferm.

Paramtre d'appel du service : `service=@merge`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Ce champ est obligatoire pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.

Paramtres en sortie

Paramtre	Libell	Description
`mergeResult`	Russite de la fusion	Contient 'true' si la fusion a russi, 'false' sinon.
`mergeState`	tat de la fusion	Contient le code de retour de la fusion. Il est fortement recommand de parser cette valeur en utilisant le classe UIHttpManagerComponentReturnCode.

37.11 Accder l'interface de fusion des espace de donnes

Le service merge.view est automatiquement considr comme termin.

Paramtre d'appel du service : `service=@merge.view`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Ce champ est obligatoire pour ce service.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.

37.12 **Comparer deux contenus**

Un workflow considre automatiquement le service compare comme termin.

Paramtre d'appel du service : `service=@compare`

Paramtres en entre

Paramtre	Libell	Description
`branch`	Espace de donnes	Identifiant de l'espace de donnes concern - Un espace de donnes ou une image, et un espace de donnes ou une image comparer sont obligatoires pour ce service.
`compare.branch`	Espace de donnes comparer	Identifiant de l'espace de donnes comparer - Un espace de donnes et un espace de donnes comparer sont obligatoires pour ce service.
`compare.filter`	Filtre de comparaison	Pour ignorer l'hritage et les champs de valeurs calcules lors de la comparaison (dsactiver le mode rsolu), le filtre "persistedValuesOnly" doit tre spcifi. Par dfaut, lorsqu'aucun filtre n'est dfini, la comparaison utilise le mode rsolu.
`compare.instance`	Jeu de donnes comparer	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes comparer saisi.
`compare.version`	Image comparer	Identifiant de l'image comparer - Un espace de donnes et un espace de donnes comparer sont obligatoires pour ce service.
`compare.xpath`	Table ou enregistrement comparer (XPath)	La valeur saisie doit tre un chemin absolu d'une table ou d'un enregistrement dans le jeu de donnes comparer saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge.
`instance`	Jeu de donnes	La valeur saisie doit tre une rfrence vers un jeu de donnes qui existe dans l'espace de donnes saisi.
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service, c'est--dire les entits que l'utilisateur peut slectionner dans sa session. Si non renseigne, la valeur utilise sera celle par dfaut. Pour les perspectives, la valeur par dfaut est toujours 'node'. Pour les workflows, la valeur par dfaut dpend des entits ou du service slectionns.
`trackingInfo`	Informations de suivi	Les informations de suivi sont enregistres dans les logs 'historique'. Ces informations peuvent galement tre utilises dans d'autres buts comme le contrle d'accs ou une mise disposition d'informations supplmentaires.
`version`	Image	Identifiant de l'image concerne - Un espace de donnes ou une image, et un espace de donnes ou une image

Paramtre	Libell	Description
		comparer sont obligatoires pour ce service.
`xpath`	Table ou enregistrement (XPath)	La valeur saisie doit tre un chemin absolu d'une table ou d'un enregistrement dans le jeu de donnes saisi. La notation doit tre conforme la convention XPath, sous sa forme abrge.

37.13 **Workflows de donnes**

Ce service fournit un accs aux interfaces utilisateurs des workflows de donnes.

Paramtre d'appel du service : `service=@workflow`

> **Note**
>
> Ce service est destin aux perspectives uniquement.

Paramtres en entre

Paramtre	Libell	Description
`scope`	Primtre	Dfinit le primtre de la navigation utilisateur pour ce service.
`viewPublication`	Publication de vue	Dfinit le nom de publication de la vue appliquer pour ce service.
`workflowView`	Type d'affichage	Spcifie le type d'affichage du workflow. La valeur peut tre une des suivantes: "inbox" (boîte de rception), "launcher" (lanceurs de workflows), "monitoringPublications" (publications), "monitoringWorkflows" (workflows actifs), "monitoringWorkItems" (bons de travail) ou "completedWorkflows" (workflows termins).
`xpath`	Filtre (XPath)	Un paramtre optionnel permettant de filtrer les donnes affiches. La syntaxe doit respecter celle d'un prdicat XPath entour par "[" et "]".

CHAPITRE **38**

Import et export XML

Ce chapitre contient les sections suivantes :

1. Introduction
2. Imports
3. Exports
4. Gestion des valeurs de champ
5. Limitations connues

38.1 Introduction

L'import et l'export XML des tables s'effectuent via l'interface utilisateur partir du menu 'Actions' de l'espace de travail.

L'import et l'export sont raliss dans le contexte d'un jeu de donnes.

L'import et l'export peuvent aussi tre raliss programmatiquement.

Les valeurs par dfaut des options peuvent tre dfinies dans 'Administration', sous *Interface utilisateur > Configuration de l'interface graphique > Valeurs par dfaut des options > Import / Export*.

38.2 Imports

Attention

Les documents XML imports doivent tre encods selon la norme UTF-8 et leur structure doit respecter le modle de donnes du jeu de donnes cible.

Mode d'import

Lors de l'import d'un fichier XML, il est ncessaire de spcifier un des modes d'import suivants, qui dterminera la faon dont la procdure d'import gre les enregistrements source.

Insertion seulement	Seule la cration d'enregistrement est autorise. Si un enregistrement avec la mme cl primaire existe dans la table, une erreur se produit et l'opration est annule.
Mise jour seulement	Seule la mise jour d'enregistrement est autorise. Si un enregistrement avec la mme cl primaire n'existe pas dans la table, une erreur se produit et l'opration est annule.
Mise jour ou insertion	Si un enregistrement avec la mme cl primaire existe dans la table, il est mis jour;sinon, il est cr.
Remplacement (synchronisation)	Si un enregistrement avec la mme cl primaire existe dans la table cible, celui-ci est mis jour;sinon, un nouvel enregistrement est cr. D'autre part, si un enregistrement n'est plus prsent dans la source, il est supprim.

Oprations d'insertion et de mise jour

Le mode '*by delta*' permet d'ignorer les lments du modle de donnes qui manquent dans le document XML source. Ce mode peut tre activ via les services de donnes ou l'API Java. Le tableau suivant rsume le comportement des oprations d'insertion et de mise jour lorsque les lments sont absents du document source.

Voir les oprations des services de donnes mise jour [p 647] et insertion [p 649], ainsi que `ImportSpec.setByDelta`[API] dans l'API Java pour plus d'informations.

tat dans le document XML source	Comportement
L'lment n'existe pas dans le document source	**Si le mode 'by delta' est dsactiv (par dfaut) :** Le champ cible prend une des valeurs suivantes : • Si l'lment dfinit une valeur par dfaut, le champ cible prend cette valeur par dfaut. • Si l'lment est d'un type autre que chaîne de caractres ou liste, le champ cible prend la valeur `null`. • Si l'lment est une liste agrge, la valeur du champ cible prend la valeur d'une liste vide. • Si l'lment est une chaîne qui diffrencie `null` d'une chaîne de caractres vide, la valeur du champ cible prend la valeur `null`. S'il s'agit d'une chaîne qui ne fait pas la diffrence entre les deux, une chaîne vide. • Si l'lment (simple ou complexe) est cach dans services de donnes, la valeur cible reste inchange. **Voir aussi***Hiding a field in Data Services* *[p 564]* **Note :** L'utilisateur qui excute l'import doit avoir les permissions ncessaires pour crer ou modifier la valeur du champ cible. Autrement, la valeur restera inchange. **Si le mode 'by delta' a t activ au travers des services de donnes ou de l'API Java :** • Pour l'opration `update`, la valeur de champ reste inchange. • Pour l'opration `insert`, le comportement est le mme que lorsque le mode `byDelta` est dsactiv.
L'lment existe tout en tant vide (par exemple, `<fieldA/>`)	• Pour des nœuds de type `xs:string` (ou un de ses sous-types), le champ cible prend la valeur `null` s'il distingue `null` d'une chaîne vide. Autrement, la valeur est une chaîne vide. • Pour les types de nœuds non`xs:string`, une exception est lance conformment au XML Schema. **Voir aussi***TIBCO EBX whitespace management for data types* *[p 549]*
L'lment est prsent et de valeur `null` (par exemple, `<fieldA xsi:nil="true"/>`)	Le champ cible prend toujours la valeur `null` sauf dans le cas des listes, pour lesquelles il n'est pas support. Afin d'utiliser l'attribut `xsi:nil="true"`, il est ncessaire d'ajouter la dclaration du namespace `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`.

Met les valeurs manquantes nul

Lors d'une mise jour d'enregistrement existant, si un nœud est absent ou vide dans le fichier XML:si cette option est "oui", il sera considr comme nul. Si cette option est "non", il ne sera pas modifi.

Ignorer les colonnes supplmentaires

Il peut arriver que le document XML contienne des lments qui n'existent pas dans le modle de donnes cible. Par dfaut, dans ce genre de cas, la procdure d'import chouera. Cependant, il est possible d'autoriser les utilisateurs lancer des procdures d'import qui ignoreront les colonnes supplmentaires dfinies dans les fichiers XML. Cela peut se dfinir dans les paramtres de configuration de l'assistant d'import XML. La valeur par dfaut de ce paramtre peut tre modifie dans la configuration de l''Interface utilisateur' dans l'espace 'Administration'.

Verrouillage optimiste

Si l'attribut technique `ebxd:lastTime` existe dans le fichier XML source, le mcanisme d'import ralise une vrification afin d'empcher une opration de mise jour sur un enregistrement qui pourrait avoir chang depuis la dernire lecture. Afin d'utiliser l'attribut `ebxd:lastTime`, i l est ncessaire d'ajouter la dclaration du namespace `xmlns:ebxd="urn:ebx-schemas:deployment_1.0"`. L'horodatage associ l'enregistrement courant sera compar cet horodatage. S'ils sont diffrents, la mise jour est rejete.

38.3 **Exports**

> **Note**
>
> Les documents XML exports sont toujours encods en UTF-8.

Lors d'un export au format XML, si des filtres sont appliqus la table, seuls les enregistrements correspondant au filtre seront inclus.

Les options d'export XML sont les suivantes :

Nom du fichier de tlchargement	Spcifie le nom du fichier XML exporter. Ce champ est pr-rempli avec le nom de la table source des enregistrements.
Mode convivial	Indique si les valeurs doivent tre prsentes de faon conviviale pour l'utilisateur ou sous leur forme brute (format XML standard). En mode convivial, les dates et les nombres sont formats selon la rgion de l'utilisateur, les cls trangres et valeurs numres prsentent les libells associs, etc. **Note:** Si cette option est slectionne, le fichier export ne pourra pas tre r-import.
Inclure les donnes techniques	Indique si des donnes techniques internes seront incluses dans l'export. **Note:** Si cette option est slectionne, le fichier export ne pourra pas tre r-import.
Indent	Spcifie si le fichier doit tre indent pour amliorer sa lisibilit par un humain.
Enlever le commentaire XML	Spcifie si le commentaire XML gnr qui dcrit la localisation des donnes et la date d'export doit tre enlev.

38.4 Gestion des valeurs de champ

Date, heure & format dateTime

Les formats de date et d'heure suivants sont supports :

Type	Format	Exemple
xs:date	aaaa-MM-jj	2007-12-31
xs:time	HH:mm:ss ou HH:mm:ss.SSS	11:55:00
xs:dateTime	aaaa-MM-jjTHH:mm:ss ou aaaa-MM-jjTHH:mm:ss.SSS	2007-12-31T11:55:00

38.5 Limitations connues

Champs d'association

Les services d'import et d'export XML ne supportent pas les valeurs d'association.

L'export de ces champs ne causera aucune erreur, cependant, aucune valeur ne sera exporte.

L'import de ces champs causera une erreur et la procdure d'import chouera.

Nœuds de slection

Les services d'import et d'export XML ne supportent pas les valeurs de slection.

L'export de ces champs ne causera aucune erreur, cependant, aucune valeur ne sera exporte.

L'import de ces champs causera une erreur et la procdure d'import chouera.

CHAPITRE 39

Import et export CSV

Ce chapitre contient les sections suivantes :

1. Introduction
2. Exports
3. Imports
4. Gestion des valeurs de champ
5. Limitations connues

39.1 Introduction

L'import et l'export CSV peuvent tre raliss sur des tables via l'interface utilisateur en utilisant le menu 'Actions' de l'espace de travail.

L'import et l'export sont raliss dans le contexte d'un jeu de donnes.

L'import et l'export peuvent aussi tre raliss programmatiquement.

Les valeurs par dfaut des options peuvent tre dfinies dans 'Administration', sous *Interface utilisateur > Configuration de l'interface graphique > Valeurs par dfaut des options > Import / Export*.

Voir aussi*Default option values* *[p 415]*

39.2 Exports

Lors d'un export au format CSV, si des filtres sont appliqus la table, seuls les enregistrements correspondant au filtre seront inclus.

Les options d'export CSV sont :

Nom du fichier de tlchargement	Spcifie le nom du fichier CSV exporter. Ce champ est pr-rempli avec le nom de la table source des enregistrements.
Jeu de caractres	Spcifie le jeu de caractre utiliser pour le fichier export. La valeur par dfaut est UTF-8.
Activer l'hritage	Afin de prendre en compte l'hritage [p 27] lors d'un export CSV, l'option doit tre spcifie au pralable dans le modle. Pour plus d'informations sur l'hritage, voir Hritage et rsolution de valeur [p 294]. Spcifie si l'hritage est pris en compte durant l'export CSV. Si l'hritage est activ, les valeurs de champs rsolues sont exportes avec les donnes techniques qui dfinissent le mode d'hritage potentiel de l'enregistrement ou du champ. Si l'hritage est dsactiv, les valeurs de champs rsolues sont exportes et les enregistrements occults sont ignors. Par dfaut, cette option est dsactive. **Note:**L'hritage est toujours ignor lorsque le jeu de donnes de la table n'a pas de parent ou si la table n'a pas de champ hrit.
Mode convivial	Indique si les valeurs doivent tre prsentes de faon conviviale pour l'utilisateur ou sous leur forme brute (format XML standard). En mode convivial, les dates et les nombres sont formats selon la rgion de l'utilisateur, les cls trangres et valeurs numres prsentent les libells associs, etc. **Note :** Si cette option est slectionne, le fichier export ne pourra pas tre r-import.
Inclure les donnes techniques	Indique si des donnes techniques internes seront incluses dans l'export. **Note :** Si cette option est slectionne, le fichier export ne pourra pas tre r-import.
En-ttes de colonne	Spcifie s'il faut ou non inclure les en-ttes de colonne dans le fichier CSV. • **Pas d'en-tte** • **Libell :** Une ligne est ajoute au dbut du fichier CSV et contient dans chaque colonne son libell correspondant. Chaque libell est localis conformment la prfrence de langue de la session active. Si aucun libell n'est dfini pour un nœud, le nom technique du nœud est utilis.

	• **XPath:** Une ligne est ajoute au dbut du fichier CSV et contient dans chaque colonne le chemin d'accs correspondant.
Sparateur de champ	Spcifie le sparateur de champ utiliser lors des exports. Le sparateur par dfaut est la virgule, il peut tre redfini dans *Administration > Interface utilisateur*.
Sparateur de liste	Spcifie le sparateur utiliser pour les listes de valeurs. Le sparateur par dfaut est le retour la ligne, il peut tre redfini dans *Administration > Interface utilisateur*.

Les exports CSV programmatiques sont raliss en utilisant les classes `ExportSpec`[API] et `ExportImportCSVSpec`[API] dans l'API Java.

39.3 **Imports**

Nom du fichier de tlchargement	Spcifie le nom du fichier CSV importer.
Mode d'import	Lors de l'import d'un fichier CSV, il est ncessaire de spcifier un des modes suivants, qui contrlera l'intgrit des oprations entre la source et la table cible. • **Insertion seulement :** Seule la cration d'enregistrement est autorise. Si un enregistrement avec la mme cl primaire existe dans la table, une erreur se produit et l'opration est annule. • **Mise jour seulement :** Seule la mise jour d'enregistrement est autorise. Si un enregistrement avec la mme cl primaire n'existe pas dans la table, une erreur se produit et l'opration est annule. • **Mise jour ou insertion :** Si un enregistrement avec la mme cl primaire existe dans la table, il est mis jour;sinon, il est cr. • **Remplacement (synchronisation) :** Si un enregistrement avec la mme cl primaire existe dans la table cible, celui-ci est mis jour;sinon, un nouvel enregistrement est cr. D'autre part, si un enregistrement n'est plus prsent dans la source, il est supprim.
Jeu de caractres	Spcifie le jeu de caractre utiliser pour le fichier import. La valeur par dfaut est UTF-8.
En-ttes de colonne	Spcifie s'il faut ou non inclure les en-ttes de colonne dans le fichier CSV. • **Pas d'en-tte** • **Libell :** Une ligne est ajoute au dbut du fichier CSV et contient dans chaque colonne son libell correspondant. Chaque libell est localis conformment la prfrence de langue de la session active. Si aucun libell n'est dfini pour un nœud, le nom technique du nœud est utilis. • **XPath:** Une ligne est ajoute au dbut du fichier CSV et contient dans chaque colonne le chemin d'accs correspondant.
Sparateur de champ	Spcifie le sparateur de champ utiliser lors des imports. Le sparateur par dfaut est la virgule, il peut tre redfini dans *Administration > Interface utilisateur*.

Sparateur de liste	Spcifie le sparateur utiliser pour les listes de valeurs. Le sparateur par dfaut est le retour la ligne, il peut tre redfini dans *Administration > Interface utilisateur*.
Activer l'hritage	Afin de prendre en compte l'hritage [p 27] lors d'un export CSV, l'option doit tre spcifie au pralable dans le modle. Pour plus d'informations sur l'hritage, voir Hritage et rsolution de valeur [p 294] et `ExportImportCSVSpec.setInheritanceEnabled`[API]. Spcifie si l'hritage est pris en compte pendant un import CSV. Si les donnes techniques dans le fichier CSV dfinissent un mode d'hritage, les champs ou les enregistrements correspondants sont obligatoirement hrits. Si des donnes techniques dfinissent un mode occult, les enregistrements correspondants sont obligatoirement occults. Autrement, les champs sont crass au profit de valeurs issues du fichier CSV. Par dfaut, cette option est dsactive. **Note :** L'hritage est toujours ignor lorsque le jeu de donnes de la table n'a pas de parent ou si la table n'a pas de champ hrit.

Les imports CSV programmatiques sont raliss en utilisant les classes `ImportSpec`[API] et `ExportImportCSVSpec`[API] dans l'API Java.

39.4 Gestion des valeurs de champ

Listes agrges

Les services d'import et d'export CSV supportent les champs valeurs multiples, savoir les listes agrges. Seules les listes simples telles que les listes de `string`, `date`, ou `int` et les cls trangres sont supportes. Si une cl trangre est lie un enregistrement via une cl primaire compose, chaque champ de la cl trangre est une chaîne formate, par exemple, "true|99". Les listes agrges des groupes ne sont pas exportes.

Lors d'un export, les lments de la liste sont spars par le sparateur de ligne. Dans les cas o le champ export contient dj un sparateur de ligne, par exemple dans un `osd:html` ou un `osd:text`, le code `_crnl_` est insr la place du sparateur de ligne du champ de valeur. Le mme formatage est attendu lors de l'import, l'ensemble des valeurs de champ sont alors entoures de guillemets.

Champs cachs

Les champs cachs sont exports en tant que chaînes `ebx-csv:hidden`. Une chaîne cache importe ne modifiera pas le contenu d'un champ.

Valeur de chaîne 'Null'

En utilisant les services d'import et d'export CSV, une chaîne dont la valeur est `null` est exporte en tant que chaîne vide. Par consquent, une boucle de procdure d'export-import finira par remplacer les valeurs de chaîne `null` par des chaînes vides.

En utilisant les services programmatiques, la valeur spcifique `ebx-csv:nil` peut tre assigne des chaînes dont la valeur est `null`. Dans ce cas, les valeurs de chaîne `null` ne seront pas remplaces par des chaînes vides lors de boucles de procdure export-import. Voir `ExportImportCSVSpec.setNullStringEncoded`[API] dans l'API Java pour plus d'informations.

Formats de date, heure et dateTime

Les formats de date et d'heure suivants sont supports :

Type	Format	Exemple
xs:date	aaaa-MM-jj	2007-12-31
xs:time	HH:mm:ss ou HH:mm:ss.SSS	11:55:00
xs:dateTime	aaaa-MM-jjTHH:mm:ss ou aaaa-MM-jjTHH:mm:ss.SSS	2007-12-31T11:55:00

39.5 Limitations connues

Listes agrges de groupes

Les services d'import et d'export CSV ne supportent pas les groupes valeurs multiples, savoir, des listes agrges d'lments de type complexe. L'export de ces nœuds ne causera aucune erreur, cependant, aucune valeur ne sera exporte.

Groupes terminaux

Dans un fichier CSV, il est impossible de diffrencier un groupe terminal cr, contenant uniquement des champs vides, d'un groupe terminal non cr.

Par consquent, quelques diffrences peuvent apparaître lors de la comparaison aprs avoir ralis une boucle de procdure d'export-import. Afin de s'assurer de la symtrie de l'import et de l'export, il est prfrable d'utiliser la fonction import et export XML. Voir Import et export XML [p 235].

En-ttes de libell de colonne

Si deux colonnes partagent le mme libell, l'export de la table peut tre ralis avec succs, mais les donnes exportes ne pourront pas tre r-importes par la suite.

Champs d'association

Les services d'import et d'export CSV ne supportent pas les valeurs d'association, c'est dire les enregistrements associs.

L'export de ces champs ne causera aucune erreur, cependant, aucune valeur ne sera exporte.

L'import de ces champs causera une erreur et la procdure d'import chouera.

Nœuds de slection

Les services d'import et d'export CSV ne supportent pas les valeurs de slection, c'est dire les enregistrements slectionns.

L'export de ces champs ne causera aucune erreur, cependant, aucune valeur ne sera exporte.

L'import de ces champs causera une erreur et la procdure d'import chouera.

CHAPITRE **40**

Syntaxe XPath supporte

Ce chapitre contient les sections suivantes :

1. Vue d'ensemble
2. Exemple d'expressions
3. Spcifications de syntaxe pour les expressions XPath
4. API Java

40.1 Vue d'ensemble

L'criture XPath utilise dans TIBCO EBX doit tre conforme la *syntaxe abrge* du XML Path Language (XPath) Version 1.0 standard, avec certaines restrictions. Ce document dcrit la syntaxe abrge supporte.

40.2 Exemple d'expressions

L'expression gnrale XPath est :

```
path[predicate]
```

Chemin absolu

```
/bibliotheque/livres/
```

Chemins relatifs

```
./Auteur
../Titre
```

Racine et chemins descendants

```
//livres
```

Chemins de table avec prdicats

```
../../livre/[auteur_id = 0101 and (editeur = 'harmattan')]
/bibliotheque/livres/[not(editeur = 'dumesnil')]
```

Prdicats complexes

```
starts-with(col3,'xxx') and ends-with(col3,'yyy') and osd:is-not-null(./col3))
contains(col3 ,'xxx') and ( not(col1=100) and date-greater-than(col2,'2007-12-30') )
```

Prdicats avec paramtres

```
author_id = $param1 and publisher = $param2
```
o les paramtres $param1 et $param2 ont respectivement pour valeurs 0101 et 'harmattan'

```
col1 < $param1 and col4 = $param2
```
o les paramtres $param1 et $param2 ont respectivement pour valeurs 100 et 'true'

```
contains(col3,$param1) and date-greater-than(col2,$param2)
```
o les paramtres $param1 et $param2 ont respectivement pour valeurs 'xxx' et '2007-12-30'

> **Note**
>
> L'utilisation de cette notation est uniquement limite l'API Java puisque les valeurs des paramtres peuvent uniquement tre dfinies par la mthode `Request.setXPathParameter`[API].

Prdicats sur libell

```
osd:label(./delivery_date)='12/30/2014' and ends-with(osd:label(../adress),'Beijing - China')
```

Prdicats sur libell d'un enregistrement

```
osd:contains-record-label('dumesnil') or osd:contains-record-label('harmattan')
```

Prdicats pour la recherche sur validation

```
osd:has-validation-item()
osd:has-validation-item('error,info')
osd:contains-validation-message('xxx')
osd:contains-validation-message('xxx','info,warning')
```

> **Note**

- Les fonctions XPath de validation sont interdites pour les associations et pour les filtres de cl trangre.
- Les prdicats `osd:label`, `osd:contains-record-label` et `osd:contains-validation-message` sont localiss. La locale peut tre dfinie par l'une des mthodes de l'API Java `Request.setLocale`[API] ou `Request.setSession`[API].

> **Attention**
>
> Afin de garantir que le rapport de validation est jour lors de la recherche, il est ncessaire de valider explicitement la table juste avant d'utiliser ces prdicats.

40.3 Spcifications de syntaxe pour les expressions XPath

Vue d'ensemble

Expression	Format	Exemple
Expression XPath	<*chemin container*>[*predicate*]	`/livres[titre='xxx']`
<*container path*>	<*chemin absolu*> ou <*chemin relatif*>	
<*absolute path*>	`/a/b` ou `//b`	`//livres`
<*relative path*>	`../../b`, `./b` ou `b`	`../../livres`

Spcification du prdicat

Expression	Format	Notes/Exemple
<*predicate*>	Exemple:A et (B et non(C)) A,B,C:<*expression atomique*>	Composition de:parenthses d'oprateurs logiques, not() et expressions atomiques.

<*atomic expression*>	<*chemin*><*comparateur*><*critre*> ou mthode(<*chemin*>,<*critre*>)	`royalty = 24.5`
		`starts-with(title, 'Johnat')`
		`booleanValue = true`

<*path*>	*<chemin relatif> ou osd:label(<chemin relatif>)*	Relatif la table qui le contient : `../authorstitle`
<*comparator*>	*<comparateur boolen>*, *<comparateur numrique>* ou *<comparateur 'string'>*	
<*boolean comparator*>	`=` ou `!=`	
<*numeric comparator*>	`=` ,`!=` ,<, >, <=, ou >=	
<*string comparator*>	`=`	
<*method*>	*<mthode date>*, *<mthode 'string'>*, mthode `osd:is-null` ou mthode `osd:is-not-null`	
<*date, time & dateTime method*>	`date-less-than`, `date-equal` ou `date-greater-than`	
<*string method*>	`matches`, `starts-with`, `ends-with`, `contains`, `osd:is-empty`, `osd:is-not-empty`, `osd:is-empty-or-nil`, `osd:is-neither-empty-nor-nil`, `osd:is-equal-case-insensitive`, `osd:starts-with-case-insensitive`, `osd:ends-with-case-insensitive`, `osd:contains-case-insensitive`, ou `osd:contains-record-label`	
<*criterion*>	*<critre boolen>*, *<critre numrique>*, *<critre 'string'>*, *<critre date>*, *<critre 'time'>*, ou *<critre 'dateTime'>*	
<*boolean criterion*>	`true` ,`false`	
<*numeric criterion*>	Un entier ou un dcimal	`-4.6`
<*string criterion*>	Chaîne de caractres entre apostrophes	`'azerty'`

Expression	Format	Notes/Exemple
<*date criterion*>	Entre apostrophes et format comme suit 'aaaa-MM-jj'	`'2007-12-31'`
<*time criterion*>	Entre apostrophes et format comme suit 'HH:mm:ss' ou 'HH:mm:ss.SSS'	`'11:55:00'`
<*dateTime criterion*>	Entre apostrophes et format comme suit 'aaaa-MM-jjTHH:mm:ss' ou 'aaaa-MM-jjTHH:mm:ss.SSS'	`'2007-12-31T11:55:00'`

Formule XPath 1.0

Il est possible d'utiliser une formule XPath 1.0 dans la partie valeur du critre d'une expression de prdicat atomique (ct droit).

Par exemple, au lieu de `[./a=3]`, il est possible d'utiliser l'expression `[./a=(floor(./d)+ 2.0)]`.

En raison de la forte dpendance des prdicats et le type du noeud du critre, la portion de chemin de l'expression du prdicat atomique (ct gauche) doit tre un chemin de noeud et ne peut tre une formule XPath. Par exemple, l'expression `/table[floor(./a) > ceiling(./d)]` n'est pas valide.

Prdicat sur libell

La fonction `osd:label()` peut tre applique la portion de chemin du prdicat atomique, afin de rsoudre le prdicat sur le libell et non sur la valeur. Dans ce cas, seuls les oprateurs et les critres de chaîne peuvent tre utiliss, i.e. `ends-with(osd:label(./price),'99')`.

Un prdicat sur libell est localis, le critre doit donc tre exprim dans la mme locale que la requte filtre par prdicat. Par exemple:`request.setLocale(Locale.FRENCH); request.setXPathFilter("osd:label(./delivery_date)='30/12/2014'");`

> **Note**
>
> Il est interdit d'utiliser la fonction `osd:label` si la partie droite du prdicat est une valeur contextuelle.

> **Note**
>
> Si la fonction `osd:label` est utilise dans un modle de donnes, par exemple dans une slection ou dans le prdicat de filtre du noeud de rfrence d'une table, la locale par dfaut du modle de donnes (telle que dfinie dans sa dclaration de module) doit tre utilise pour le format du critre (mme si cela n'est gnralement pas recommand).

Voir aussi *`SchemaNode.displayOccurrence`*[API]

Valeurs contextuelles

Pour les prdicats se rapportant un noeud de slection, la valeur du critre (savoir, le cot droit du prdicat) peut tre remplace par un chemin contextuel utilisant la syntaxe `${<relative-path>}` o <*relative-path*> est l'emplacement de l'lment li au noeud de slection.

> **Note**
>
> Lors de l'appel une mthode, le critre est le deuxime paramtre, et le premier paramtre ne peut pas tre une valeur relative.

Listes groupes

Pour les prdicats sur listes groupes, le prdicat renvoie `true` indpendamment du comparateur si l'un des lments de la liste vrifie le prdicat.

> **Note**
>
> Une attention particulire doit tre porte au comparateur `!=`. Par exemple, pour une liste groupe, `./list != 'a'` est diffrent de `not(./list = 'a')`. L o la liste contient les lments `(e1,e2,..)`, le premier prdicat quivaut `e1 != 'a' or e2 != 'a' ...`, alors que le second quivaut `e1 != 'a' and e2 != 'a' ....`

Valeurs 'Null'

Les valeurs Null doivent tre traites explicitement dans uns prdicat utilisant les oprateurs `osd:is-null` et `osd:is-not-null`.

Par exemple, `/root/products[./price<100]` ou `/root/products[./price!=100]` ne retournera aucun produit dont les prix ne sont pas dfinis (`null`). Dans ce cas, pour retourner galement les valeurs non dfinies, le prdicat devrait plutt tre:`/root/products[./price!=100 or osd:is-null(./price)]`.

Comment grer les apostrophes simples et doubles dans les expressions littrales

Par dfaut, une expression littrale est dlimite par des apostrophes simples (`'`). Si l'expression littrale contient des apostrophes simples et pas d'apostrophes doubles, l'expression doit tre dlimite par des apostrophes doubles (`"`). Si l'expression littrale contient des apostrophes simples et doubles, les apostrophes simples doivent tre doubles.

Gr par la mthode `XPathExpressionHelper.encodeLiteralStringWithDelimiters`[API] dans l'API Java.

Exemples d'utilisation de `encodeLiteralStringWithDelimiters`

Valeur de l'expression littrale	**Rsultat de cette mthode**
`Coeur`	`'Coeur'`
`Coeur d'Alene`	`"Coeur d'Alene"`
`Il a dit: "Ils vivent dans Coeur d'Alene".`	`'Il a dit: "Ils vivent dans Coeur d''Alene".'`

Extraction de cls trangres

Dans EBX, les cls trangres sont regroupes dans un champ unique avec la dclaration `osd:tableRef` [p 522].

La syntaxe XPath standard a t tendue afin d'extraire la valeur de n'importe quel champ de cl primaire cibl.

Exemple

Si la table `/root/tableA` a un champ `osd:tableRef` nomm 'fkB' dont la cible est `/root/tableB` et la cl primaire de `tableB` a deux champs, `id` de type `xs:int` et `date` de type `xs:date`, alors les expressions suivantes sont valides :

- `/root/tableB[ fkB = '123|2008-01-21' ]`, o la chaîne "123|2008-01-21" est une reprsentation de l'intgralit de la valeur de la cl primaire.

  Voir **Syntaxe de la reprsentation de la chaîne interne des cls primaires** `PrimaryKey.syntax`[API] pour plus d'informations.
- `/root/tableA[ fkB/id = 123 and date-equal(fkB/date, '2008-01-21') ]`, o ce prdicat est un quivalent plus efficace que celui de l'exemple prcdent.
- `/root/tableA[ fkB/id >= 123 ]`, o n'importe quel oprateur numrique pourrait tre utilis, puisque le champ de cl primaire cible est de type `xs:int`.
- `/root/tableA[ date-greater-than( ./fkB/date,'2007-01-01' ) ]`, o n'importe quel oprateur date pourrait tre utilis, puisque le champ de cl primaire cible est de type `xs:date`;
- `/root/tableA[ fkB = "" ]` n'est pas valide puisque la cl primaire cible possde deux colonnes.
- `/root/tableA[ osd:is-null(fkB) ]` vrifie si une cl trangre est `null` (non dfinie).

40.4 API Java

Utilisation de XPath dans l'API Java :

En API Java, la classe `XPathFilter` permet de dfinir des prdicats XPath et d'excuter des requtes sur ceux-ci.

La classe `XPathExpressionHelper` offre des mthodes utilitaires pour manipuler les prdicats et les chemins XPath.

Localisation

CHAPITRE 41

Libells et localisation

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Rgles de formatage des valeurs
3. Syntaxe des locales

41.1 Prsentation

TIBCO EBX donne la possibilit de grer les libells et l'internationalisation des modles de donnes.

Localisation de l'interface utilisateur

Dans EBX, les prfrences de langue peuvent tres dfinies sur deux primtres distincts :

1. La session:chaque utilisateur peut slectionner une locale par dfaut partir du panneau utilisateur.
2. Le fichier principal de configuration d'EBX, intitul `ebx.properties` par dfaut. Voir Extension de l'internationalisation de TIBCO EBX [p 261] pour plus d'informations.

Informations textuelles

Dans EBX, la plupart des entits de donnes de rfrence ont un libell et une description, ou sont lies un message utilisateur. Par exemple :

- Les espaces de donnes, les images et les jeux de donnes peuvent avoir leurs propres libells et descriptions. Le libell est indpendant du nom unique, ce qui permet la traduction ainsi que la modification ;
- N'importe quel noeud du modle de donnes peut avoir un libell et une description statiques ;
- Les valeurs numres peuvent avoir un libell statique ;
- Les messages de validation peuvent tre personnaliss, et les restrictions de permission peuvent fournir un texte expliquant le motif ;
- Chaque enregistrement est affich dynamiquement en fonction de son contenu, ainsi que le contexte dans lequel il est affich (dans une hirarchie, en tant que cl trangre, etc.) ;

Toutes ces informations textuelles peuvent tre traduites dans les locales dclares dans `ebx.properties`.

Voir aussi

Libells et messages [p 555]

Dclaration de tables [p 517]

Dclaration de cls trangres [p 522]

41.2 Rgles de formatage des valeurs

Lorsqu'une valeur est affiche pour l'utilisateur, elle est formate conformment son type et la rgle de formatage de la locale actuelle. Par exemple, une date sera affiche dans certaines locales "jj/MM/aaaa" et "MM/jj/aaaa" dans d'autres.

Une rgle de formatage est utilise pour dfinir la manire dont sont affiches les valeurs des types simples [p 504].

Pour chaque locale dclare dans `ebx.properties`, sa rgle de formatage est configure dans un fichier : `/WEB-INF/ebx/{locale}/frontEndFormattingPolicy.xml`. Par exemple, pour dfinir la rgle de formatage destine au grec (`el`), le systme recherche le chemin suivant dans le module:

```
/WEB-INF/ebx/el/frontEndFormattingPolicy.xml
```

Si le fichier correspondant n'existe pas, la rgle de formatage est recherche dans le class-path de EBX. Si la rgle de formatage spcifique la locale n'est pas trouve, la rgle de formatage `en_US` sera applique.

Le contenu du fichier `frontEndFormattingPolicy.xml` est le suivant:

```
<?xml version="1.0" encoding="UTF-8"?>
<formattingPolicy  xmlns="urn:ebx-schemas:formattingPolicy_1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ebx-schemas:formattingPolicy_1.0 ../schema/ebx-reserved/formattingPolicy_1.0.xsd">
 <date pattern="dd/MM" />
 <time pattern="HH:mm:ss" />
 <dateTime pattern="dd/MM/yyyy HH:mm" />
 <decimal pattern="00,00,00.000" groupingSeparator="|" decimalSeparator="^"/>
 <int pattern="000,000" groupingSeparator=" "/>
</formattingPolicy>
```

Les lments `date`, `dateTime` et `time` sont obligatoires.

Les sparateurs de groupes et de dcimaux qui apparaissent lors du formatage des nombres peuvent tre modifis en dfinissant les attributs `groupingSeparator` et `decimalSeparator` pour les lments `decimal` et `int`.

41.3 Syntaxe des locales

Il y a deux manires d'exprimer une locale :

1. La recommandation XML suit la recommandation IETF BCP 47, qui utilise le trait d'union "-" comme sparateur.
2. La spcification Java utilise le tiret bas "_" au lieu du trait d'union.

Dans n'importe quel fichier XML (XSD, fichier de rgle de formatage, etc.) lu par EBX, les deux syntaxes sont autorises.

Pour un chemin web, savoir un chemin contenu dans l'application web, seule la syntaxe Java est autorise. Ainsi, les fichiers de rgles de formatage doivent se trouver dans des rpertoires dont le nom de locale respecte la syntaxe Java.

Voir aussi*Extension de l'internationalisation de TIBCO EBX [p 261]*

CHAPITRE 42

Extension de l'internationalisation de TIBCO EBX

Ce chapitre contient les sections suivantes :

1. Localisation native d'EBX
2. Extension de la localisation de l'interface utilisateur d'EBX
3. Rsolution des ressources localises
4. Limitations connues

42.1 Localisation native d'EBX

Par dfaut, l'interface utilisateur intgre d'EBX est fournie en anglais (en-US) et en franais (fr-FR). La localisation est constitue des rgles de formatage de donnes et d'un ensemble de fichiers de messages (fichier de ressources) :

- Pour l'anglais, la localisation est fournie par une rgle de formatage de donnes et un ensemble de fichiers de messages sans locale dfinie,
- Pour le franais, la localisation est fournie par une rgle de formatage et un ensemble de fichiers de messages avec une locale dfinie "fr".

EBX offre la possibilit d'ajouter des locales pour tendre la localisation de l'interface utilisateur et d'internationaliser la documentation des modles de donnes et services associs.

42.2 Extension de la localisation de l'interface utilisateur d'EBX

EBX prend en charge la localisation de son interface utilisateur vers n'importe quelle langue ou rgion compatible.

> **Note**
>
> Actuellement, seuls les caractres latins et cyrilliques sont accepts. Les locales utilisant d'autres jeux de caractres sont utilisables, mais ne sont pas prises en charge.

Ajouter une nouvelle locale

Pour ajouter une nouvelle locale, suivre les tapes suivantes :

- Dclarer la nouvelle locale dans le fichier principal de configuration d'EBX. Par exemple : ebx.locales.available=en-US, fr-FR, xx
  - La premire locale est toujours considre comme tant celle par dfaut.
  - Les locales intgres, en-US et fr-FR, peuvent tre supprimes si besoin.

  Voir Configuring EBX localization [p 373].
- Dployer les fichiers suivants dans le class-path d'EBX :
  - Un fichier de rgles de formatage, nomm `com.orchestranetworks.i18n.frontEndFormattingPolicy_xx.xml`,
  - Un ensemble de fichiers de messages localiss (*_xx.mxml) dans un fichier de ressources.

    > **Note**
    >
    > Les fichiers doivent terminer par ".mxml".

42.3 Rsolution des ressources localises

Depuis la version 5.7.0, les ressources localises sont rsolues sur la base de la proximit de locale, grce au mcanisme de rsolution suivant :

- resourceName + "_" + langue + "_" + pays + "_" + variante + ".mxml"
- resourceName + "_" + langue + "_" + pays + ".mxml"
- resourceName + "_" + langue + ".mxml"
- resourceName + ".mxml"

> **Note**
>
> La rsolution s'effectue au niveau du message localis. Il est ainsi possible de dfinir un ou plusieurs fichiers pour une locale n'incluant que les messages dont la localisation spcifique est souhaite.

42.4 Limitations connues

lments non extensibles

La localisation des lments suivants ne peut pas tre tendue :

- Documentation produit EBX,
- Visualiseur et diteur HTML EBX.

Persistance

CHAPITRE 43

Prsentation de la persistance

Ce chapitre dcrit la faon dont les donnes de rfrence, l'historique et les tables rpliques sont conservs. Une table peut utiliser n'importe quelle combinaison de modes de persistance des donnes de rfrence, d'historisation et de rplication.

Bien que toutes les informations conserves dans TIBCO EBX soient ensuite stockes dans la base de donnes sous-jacente en tant que tables relationnelles, le fait qu'elles se trouvent de faon accessible en dehors d'EBX dpendra du mode mapp.

> **Note**
>
> Le terme *mode mapp* [p 265] fait rfrence toute table stocke telle quelle et dont le contenu est accessible directement en base de donnes.

Ce chapitre contient les sections suivantes :

1. Persistance des donnes de rfrence gres
2. Historisation
3. Rplication
4. Mode mapp

43.1 Persistance des donnes de rfrence gres

Les donnes modlises et gouvernes par le rfrentiel EBX peuvent tre conserves dans l'un des deux modes suivants :smantique (par dfaut) ou relationnel, selon ce qui a t spcifi dans le modle de donnes sous-jacent. Des tables distinctes dfinies dans l'un ou l'autre mode peuvent coexister et collaborer dans le mme rfrentiel EBX.

Pour comparer le mode relationnel et le mode smantique, voir le chapitre Prsentation des modes [p 267]

43.2 Historisation

Les tables de donnes de rfrence peuvent activer l'historisation afin de tracer les modifications sur leurs donnes, indpendamment du fait qu'elles soient conserves en mode smantique ou relationnel et qu'elles soient rpliques ou non.

L'historique est en mode mapp, ce qui signifie qu'il peut tre consult directement dans la base de donnes sous-jacente.

Voir aussi*Historique* *[p 273]*

43.3 Rplication

La rplication permet un accs SQL direct aux tables, en faisant une copie des donnes du rfrentiel vers les tables relationnelles de rplique en base de donnes. La rplication peut tre active sur n'importe quelle table, indpendamment de son mode de persistance (smantique ou relationnel) et du fait que l'historique soit activ ou non.

Les rpliques sont conserves en mode mapp, leur but principal tant de rendre les donnes de rfrence accessibles par requte directe en dehors d'EBX.

Voir aussi*Rplication* *[p 281]*

43.4 Mode mapp

Prsentation du mode mapp

Le mode mapp fait rfrence aux cas o les tables sont conserves dans la base de donnes relationnelle sous-jacente dans un format permettant d'accder directement aux donnes, en dehors d'EBX. Les donnes de rfrence dclares en mode relationnel, l'historique et les copies de tables sont autant d'exemples de tables en mode mapp.

Tous les cas de mode mapp impliquent une modification automatique du schma de la base de donnes (les tables, les index, etc.) le cas chant, en excutant automatiquement les commandes DDL ncessaires en arrire-plan. Ce type de procdure est toujours dclench au moment de la compilation du modle de donnes, et le rapport de compilation du modle de donnes signale toute erreur ventuelle.

Autre remarque d'ordre gnral concernant les modes mapps:la plupart du temps, lorsqu'une entit d'un modle de donnes est supprime, l'objet correspondant en base de donnes n'est pas supprim immdiatement. Il est not comme dsactiv, ce qui laisse la possibilit de ractiver l'objet par la suite. Afin de supprimer dfinitivement de la base de donnes l'objet et les donnes et ressources qui y sont associes, il faut le marquer pour la purge. La suppression a alors lieu lors de la purge gnrale suivante.

Voir aussi

Database mapping administration *[p 429]*

volutions du modle de donnes *[p 287]*

Restrictions sur le modle de donnes suite au mode mapp

En raison de la nature de la persistance directe en base de donnes sous-jacente, certaines restrictions s'appliquent toutes les tables stockes en mode mapp :

- Limitations of supported databases [p 337]
- Longueur illimite pour les chaînes de caractres:tous les champs chaînes de caractre, except les cls trangres, de type `xs:string`, leurs types drivs et `xs:anyURI`, doivent dfinir une facette 'maxLength' ou 'length'. Puisqu'un champ de cl trangre est compos des champs de la cl primaire finale de ses tables cibles, ce critre sur les facettes s'applique chacun des champs de cl primaire finale, et non au champ de cl trangre lui-mme. De plus, les limitations de la base de donnes sous-jacente concernant la longueur maximale de ses types de caractres s'appliquent, telles que *VARCHAR* et *NVARCHAR2*.

- Les longues listes de colonnes sont parfois non-indexables. Exemple sur Oracle:la base de donnes exige une limite de taille maximale cumule des colonnes comprises dans un index. Pour les chaînes de caractres, cette taille dpend galement du jeu de caractres. Si le serveur de base de donnes choue lors de la cration de l'index, il sera ncessaire d'envisager de revoir la conception des index, typiquement en rduisant la longueur des colonnes concernes, ou en incluant moins de colonnes dans l'index. Le raisonnement est qu'un index menant cette situation aurait des en-ttes tellement grands qu'il ne pourrait de toute faon pas tre efficace.
- Les champs de type `type="osd:password"` sont ignors.
- Les types complexes terminaux sont supports;cependant, ils ne peuvent tre dfinis `null` de manire globale, au niveau de l'enregistrement.

D'une faon plus gnrale, les tables en mode mapp sont soumises aux limitations du SGBDR sous-jacent. Par exemple, le nombre maximum de colonnes dans une table doit tre respect (1000 pour Oracle 11g R2, 1600 pour PostgreSQL). Remarque:une table d'historique contient deux fois plus de champs que ceux dclars dans le schma (un champ fonctionnel, plus un champ gnr pour le code de l'opration).

Les volutions du modle de donnes peuvent aussi tre limites par le SGBDR sous-jacent, en fonction du modle de donnes existant.

Voir aussi*volutions du modle de donnes* *[p 287]*

CHAPITRE 44

Mode relationnel

Ce chapitre contient les sections suivantes :

1. Prsentation des modes
2. Activation du mode relationnel pour un modle de donnes
3. Validation
4. Accs SQL aux donnes en mode relationnel
5. Limitations du mode relationnel

44.1 Prsentation des modes

Le mode smantique en dtail

Le mode smantique prsente toutes les fonctionnalits avances de TIBCO EBX en matire de gestion de donnes de rfrence. En particulier les espaces de donnes, l'hritage des jeux de donnes et les champs hrits.

Le mode smantique est le mode par dfaut de persistance des donnes gouvernes par le rfrentiel EBX. Les modles de donnes sont en mode smantique, moins que le mode relationnel [p 268] ne soit spcifi [p 269] de manire explicite.

En interne, les donnes de rfrence gres en mode smantique sont reprsentes en XML standard, conformment au document XML Schema de son modle de donnes. La reprsentation XML est compresse nouveau et segmente pour tre stocke dans des tables gnriques de bases de donnes relationnelles. Ce mode permet de stocker et d'accder efficacement aux donnes, notamment pour :

- les espaces de donnes: lors de la cration d'un espace de donnes enfant, aucune donne n'est duplique, et
- l'hritage:aucune donne n'est duplique lors de la cration d'une instance hrite.

Le mode smantique permet aussi de maintenir un nombre illimit de jeux de donnes pour chaque modle de donnes, organiss dans un nombre illimit d'espaces de donnes et d'images. Tout cela sans impacter le schma de base de donnes.

Comme ce mode n'utilise que des tables communes internes et gnriques, les modifications de structure du modle de donnes n'ont pas non plus d'impact sur le schma de base de donnes. Les volutions du modle de donnes n'impactent que le contenu des tables gnriques de la base de donnes.

Voir aussi

Services de donnes [p 98]

dataset inheritance *[p 295]*

inherited fields *[p 296]*

Le mode relationnel en dtail

Le mode relationnel, qui est un mode mapp, conserve les donnes de rfrence directement dans la base de donnes. Le but premier du mode relationnel est de pouvoir tirer parti des performances et des capacits de monte en charge de la base de donnes relationnelle sous-jacente. Cependant, le mode relationnel ne gre pas les fonctionnalits avances de gouvernance proposes en mode smantique.

Dans certains cas o les avantages de la gestion en mode smantique ne sont pas indispensables, comme pour les tables " un instant donn", ou les tables mises jour rgulirement par des systmes externes, les gains de performance offerts en mode relationnel peuvent tre plus utiles.

De manire gnrale, quand un jeu de donnes est en mode relationnel, chaque table de ce jeu de donnes possde une table correspondante dans la base de donnes, et tous les champs de son modle de donnes correspondent une colonne de table relationnelle.

Comparaison directe du mode smantique et du mode relationnel

Le tableau suivant rsume les diffrences entre les deux modes de persistance:

	Mode smantique	**Mode relationnel**
Espaces de donnes	Oui	Non
Hritage de jeu de donnes	Oui	Non
Champs hrits	Oui	Non
Modle de donnes	Toutes les fonctionnalits sont supportes.	Des restrictions s'appliquent, voir Restrictions du modle de donnes pour les tables en mode relationnel [p 272].
Lectures directes en SQL	Non	Oui, voir Lectures SQL [p 271].
critures directes en SQL	Non	Oui, mais uniquement sous certaines conditions spcifiques, voir critures SQL [p 271].
Validation des donnes	Oui, active le mode tolrant.	Oui, certaines contraintes deviennent bloquantes, voir Validation [p 269].
Transactions	Voir Concurrency and isolation levels [p 490].	Voir Concurrency and isolation levels [p 490].
volutions du modle de donnes	Voir volutions du modle de donnes [p 287] dans le Manuel de rfrence.	Voir volutions du modle de donnes [p 287] dans le Manuel de rfrence.

44.2 Activation du mode relationnel pour un modle de donnes

Le modle de donnes dclare qu'il est en mode relationnel. En raison des contraintes intrinsques au mode relationnel, telles que l'absence d'espace de donnes enfant ou d'image, un espace de données relationnel spcifique devra tre fourni, sur lequel le modle de donnes sera publi. Les espaces de données relationnels ne permettent pas de crer de sous-espaces de donnes ou d'images.

Exemple d'une dclaration de mode relationnel :

```
<xs:schema>
 <xs:annotation>
  <xs:appinfo>
   <osd:relationalMode>
    <dataSpace>aDataSpaceKey</dataSpace>
    <dataSet>aDataSetReference</dataSet>
    <tablesPrefix>aPrefixForTablesInRDBMS</tablesPrefix>
   </osd:relationalMode>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:schema>
```

avec les lments :

lment	Description	Requis
`dataSpace`	Dsigne l'espace de données dans lequel le modle de donnes doit tre publi. Cet espace de données doit tre lui-mme en mode relationnel. Aucun espace de données ou image ne peut tre cr partir de l'espace de données dclar dans ce mode.	Oui
`dataSet`	Dsigne le jeu de donnes o le modle de donnes doit tre publi.	Oui
`tablesPrefix`	Dsigne le prfixe commun utilis pour nommer les tables gnres dans la base de donnes.	Oui

44.3 Validation

Cette section dtaille l'impact du mode relationnel sur la validation de donnes.

Contraintes structurelles

Certaines contraintes du modles de donnes d'EBX vont gnrer une "contrainte structurelle" sur le schma du SGBDR sous-jacent du mode relationnel. C'est galement le cas quand l'historique de table est activ [p 273]. Les facettes suivantes sont concernes :

- les facettes `xs:maxLength` et `xs:length` sur les lments `string` ;
- les facettes `xs:totalDigits` et `xs:fractionDigits` sur les lments `xs:decimal`.

Les bases de donnes n'offrent pas le mode de validation tolrant propos par EBX. Les contraintes cites prcdemment deviennent alors des *contraintes bloquantes*. Une contrainte bloquante signifie que les mises jour non conformes seront rejetes. De plus, ces contraintes ne sont plus vrifies au cours du

processus de validation, l'exception des contraintes de cls trangres dans certains cas (voir Mode de blocage de cl trangre [p 270]). Quand une transaction n'est pas conforme une contrainte bloquante, elle est annule et une `ConstraintViolationException`[API] est leve.

Voir aussi*Contraintes bloquantes et non bloquantes [p 546]*

Mode bloquant de cl trangre

Afin de rduire le temps de validation, les contraintes de cl trangre sont automatiquement dfinies en mode bloquant si :

- Les contraintes de cl trangre sont dfinies dans une table en mode relationnel,
- Les contraintes de cl trangre sont dfinies dans une table en mode smantique ou relationnel rfrenant une table en mode relationnel.

Pour cette contrainte, le mode bloquant implique que la tentative d'excution des actions suivantes provoque une `ConstraintViolationException`[API] :

- effacer un enregistrement rfrenc par une contrainte de cl trangre,
- effacer une instance rfrence par une contrainte de cl trangre,
- fermer un espace de donnes rfrenc par une contrainte de cl trangre.

Il est cependant possible de surcharger ce comportement par dfaut en dfinissant un mode de validation spcifique. Voir Blocking and non-blocking constraints [p 546] pour plus d'informations.

Afin de garantir l'intgrit des contraintes de cl trangre aprs une criture SQL directe qui contourne le cadre de gouvernance d'EBX, les contraintes de cl trangre seront valides dans les cas suivants :

- la premire validation explicite via l'interface utilisateur ou l'API,
- la premire validation explicite via l'interface utilisateur ou l'API aprs actualisation du schma,
- la premire validation explicite via l'interface utilisateur ou l'API aprs la rinitialisation du rapport de validation d'un jeu de donnes dans l'interface utilisateur.

Important :

- L'aspect bloquant de la contrainte de cl trangre ne concerne pas les filtres qui peuvent tre dfinis. C'est dire qu'une contrainte de cl trangre est non bloquante si un enregistrement rfrenc existe mais ne rpond pas aux critres de filtre de la cl trangre. Dans ce cas, les mises jour ne sont pas rejetes et une erreur sera alors ajoute au rapport de validation.
- Les contraintes de cl trangre ne sont pas en mode bloquant lors de l'import d'archives. En effet, toutes les contraintes bloquantes, l'exception des contraintes structurelles, sont toujours dsactives lors de l'import d'archives. Ceci permet d'avoir une certaine souplesse lors de l'import d'archives lorsque, sous certaines conditions, l'import de cls trangres rfrenant des enregistrements qui n'ont pas encore t imports doit tre tolrant.

Contraintes sur l'ensemble de la table

Les **contraintes** `Constraint`[API] programmatiques sont vrifies pour chaque enregistrement de la table la validation. Si la table dfinit des millions d'enregistrements, des problmes de performance se poseront. Il est alors recommand de dfinir une **contrainte au niveau table** `ConstraintOnTable`[API].

Dans les cas o une contrainte au niveau table ne peut tre dfinie, il est recommand de dfinir au moins une **dpendance locale ou explicite** `DependenciesDefinitionContext.dependencies`[API], afin de rduire le cot de la validation incrmentale.

Voir aussi `ConstraintOnTable`[API]

44.4 Accs SQL aux donnes en mode relationnel

Cette section dcrit comment accder directement aux donnes en mode relationnel via SQL.

Voir aussi *Accs SQL l'historique* *[p 276]*

Trouver la table dans la base de donnes

Pour chaque table EBX en mode relationnel, une table correspondante est gnre dans le SGBDR. En utilisant l'interface utilisateur d'EBX, il est possible de trouver le nom de cette table en cliquant sur le panneau de documentation de la table.

Lectures SQL

Les lectures directes en SQL sont possibles sur des transactions bien gres et dure de vie courte de prfrence. Cependant, pour ce type d'accs, les permissions d'EBX ne sont pas prises en compte. Les applications autorises raliser des lectures doivent tre identifies travers d'autres processus d'authentification et permissions.

critures SQL

Les critures directes en SQL contournent le cadre de gouvernance d'EBX. Par consquent, elles doivent tre utilises avec une ***grande prcaution***. Elles peuvent tre l'origine des situations suivantes :

- chec de l'historisation des tables d'EBX ;
- chec de l'excution des triggers d'EBX ;
- chec de la vrification des permissions et des contraintes d'EBX ;
- modifications qui chappent au processus de validation incrmentale ;
- perte de visibilit des tables smantiques d'EBX, qui peuvent tre rfrences par des cls trangres.

Par consquent, les critures directes en SQL doivent tre ralises *si, et seulement si, toutes les conditions suivantes sont remplies*:

- les tables modifies ne sont pas historises et n'ont pas de triggers EBX.
- l'application ralisant les critures peut tre authentifie grce aux permissions associes, afin de garantir l'intgrit des donnes. Plus particulirement, l'intgrit des cls trangres (`osd:tableRef`) doit tre prserve tout moment. Voir Mode bloquant de cl trangre [p 270] pour plus d'informations.
- le serveur d'application qui excute EBX doit tre arrt *lorsqu'une criture est ralise*. Ceci permet de garantir que la validation incrmentale ne devienne obsolte, ce qui serait le cas dans un contexte batch.

44.5 Limitations du mode relationnel

Le mode relationnel est pleinement oprationnel mais fait l'objet de quelques limitations, listes ci-dessous. Dans le cas de l'utilisation du mode relationnel, il est fortement recommand de prendre connaissance de ces limitations et de contacter l'quipe support de TIBCO EBX https://support.tibco.com en cas de questions.

Voir Supported databases [p 337] pour les bases de donnes supportant le mode relationnel.

Restrictions du modle de donnes pour les tables en mode relationnel

Certaines restriction s'appliquent aux modles de donnes en mode relationnel :

- Restrictions sur le modle de donnes suite au mode mapp [p 265]
- Les listes agrges [p 514] ne sont pas supportes dans les tables relationnelles. Un tel schma provoquera une erreur de compilation.
- Les attributs dfinis par l'utilisateur sur des tables relationnelles provoquent des erreurs de compilation dans le modle de donnes.
- Hritage de jeu de donnes [p 295].
- Champs hrits [p 296].
- Des contraintes programmatiques, puisque le cot de calcul de la validation serait trop important. Cependant, les **contraintes sur tables** `ConstraintOnTable`[API] restent disponibles.

Les volutions de schma peuvent aussi tre limites par le SGBDR sous-jacent selon les donnes dj contenues dans les tables concernes.

Voir aussi*volutions du modle de donnes [p 287]*

Autres limitations du mode relationnel

- Limitations du mode mapp [p 265]
- Il n'est pas possible de crer d'espaces de donnes enfants ou d'images partir d'un espace de donnes contenant des jeux de donnes en mode relationnel.
- Pour D3, il n'est pas possible de diffuser un espace de donnes dfini en mode relationnel.
- Pour de trs grands volumes de donnes, la validation aura une faible performance si la table relationnelle dclare une de ces fonctionnalits:`osd:function`, `osd:select`, `osd:uiFilter`, `osd:tableRef/filter`. De plus, aucun tri ne peut tre appliqu sur une colonne `osd:function`.
- Il est impossible de dfinir `AdaptationValue.INHERIT_VALUE` sur un noeud appartenant un modle de donnes en mode relationnel.

CHAPITRE 45

Historique

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Configuration de l'historique
3. Affichage d'historique et permissions
4. Accs SQL l'historique
5. Impacts et limitations du mode historis

45.1 Prsentation

L'historique est une fonctionnalit permettant de suivre toutes les modifications effectues sur les donnes d'une table (cration d'enregistrements, mise jour ou suppression).

Ceci constitue une version amliore de la piste d'audit XML [p 443]. La piste d'audit XML est active par dfaut, mais peut tre dsactive en toute scurit si l'historique est mis en place pour les tables concernes.

Voir aussi

Historique [p 29]

Mode relationnel [p 267]

Rplication [p 281]

volutions du modle de donnes [p 287]

45.2 Configuration de l'historique

Afin d'activer l'historisation d'une table, un profil d'historique doit tre dfini pour cette table dans le modle de donnes. Cette section dcrit les profils d'historique et comment ils sont associs aux tables.

Configuration de l'historique dans le rfrentiel

Un profil d'historique spcifie le moment o l'historisation doit tre mise en place. Pour modifier les profils d'historique, slectionner *Administration > Historique et journaux*.

Un profil d'historique est identifi par un nom et dfinit les informations suivantes :

- un libell internationalis,

- une liste d'espaces de donnes (branches) pour lesquels l'historique est activ. Il est possible de spcifier si seuls les enfants directs ou si tous les descendants sont aussi concerns.

Certains profils sont dj prsents lors de l'installation du rfrentiel. Ces profils ne peuvent tre ni supprims ni modifis.

Id profil	Description
`ebx-referenceBranch`	Ce profil est uniquement activ pour l'espace de donnes de rfrence.
`ebx-allBranches`	Ce profil est activ pour tous les espaces de donnes.
`ebx-instanceHeaders`	Ce profil historise les en-ttes des jeux de donnes. Cependant, ce profil ne sera paramtrable que dans une future version, tant donn que le modle de donnes interne ne dfinit que les noeuds des jeux de donnes.

Configuration de l'historique dans le modle de donnes

Activation de l'historique de table

L'historique peut tre activ pour une table via l'assistant de modle de donnes, ou en ditant le modle de donnes sous-jacent.

Pour activer l'historique en ditant le modle de donnes, un profil d'historique doit tre dclar pour la table en utilisant l'lment `historyProfile`.

```
<osd:table>
 <primaryKeys>/key</primaryKeys>
 <historyProfile>historyProfileForProducts</historyProfile>
</osd:table>
```

L'assistant de modle de donnes permet de visualiser le profil d'historisation dfini dans le rfrentiel.

L'historisation peut tre active sparment pour chaque table. Voir la documentation de conception du modle [p 500] pour plus d'informations.

Dsactivation de l'historique d'un champ ou d'un groupe spcifique

Pour une table historise, la rgle par dfaut est l'historisation de tous les lments qu'elle prend en charge, (voir Impacts et limitations du mode historis [p 278]).

Il est possible de dsactiver l'historique d'un champ ou d'un groupe spcifique, que ce soit via l'assistant du modle de donnes ou via la modification du modle de donnes sous-jacent.

Pour dsactiver l'historique d'un champ ou d'un groupe en ditant le modle de donnes, utiliser l'lment `osd:history` avec l'attribut `disable="true"`.

```
<xs:element name="longDescription" type="xs:string">
    <xs:annotation>
        <xs:appinfo>
            <osd:history disable="true" />
        </xs:appinfo>
    </xs:annotation>
</xs:element>
```

Pour dsactiver l'historique d'un champ ou d'un groupe via l'assistant du modle de donnes, utiliser la proprit `History` dans les `Advanced properties` de l'lment.

Quand cette proprit est dfinie pour un groupe, l'historique est dsactiv rcursivement pour tous ses descendants. Lorsque l'on dsactive l'historique d'un groupe, il est impossible de ractiver spcifiquement l'historique pour un descendant.

> **Note**
>
> Si la table contenant le champ ou le groupe n'est pas historise, cette proprit n'aura aucun effet.
>
> Il n'est pas possible de dsactiver l'historique pour les champs de cl primaire.

Intgrit

Si des problmes sont dtects lors de la compilation du modle de donnes, des messages d'alerte ou d'erreur seront ajouts au rapport de validation associ ce modle de donnes. En outre, si la moindre erreur est dtecte, chaque instance (jeu de donnes) associe sera inaccessible. Les cas d'erreur les plus rpandus sont les suivants :

- Une table fait rfrence un profil qui n'est pas dfini dans le rfrentiel.
- Un profil d'historique qui est rfrenc dans le modle de donnes mentionne un espace de donnes non dfini ou ferm du rfrentiel courant.

> **Note**
>
> Le dploiement d'un modle de donnes dans un rfrentiel qui ne dispose pas des profils requis ncessitera l'intervention de l'administrateur pour les ajouter.

45.3 **Affichage d'historique et permissions**

Vue historique de table

Lorsque l'historique a t activ pour une table dans un modle de donnes, il est possible d'accder la vue de l'historique partir de l'interface utilisateur depuis:un enregistrement, une slection d'enregistrements, une table et un jeu de donnes.

La section suivante explique comment les permissions sont dtermines.

Pour plus d'informations, voir la section vue historique de table [p 29] Pour accder la vue historique de table partir de Java, la mthode `AdaptationTable.getHistory`[API] doit tre invoque.

Permissions pour historique de table

Les permissions relatives aux donnes sont aussi appliques l'historique des donnes. Les permissions de l'historique sont dfinies automatiquement, la permission la plus restrictive parmi les permissions des donnes et les droit d'accs en *lecture seule* sera applique.

C'est le cas des permissions dfinies par l'utilisateur et des rgles de permission programmatiques.

Lors de la dfinition d'une rgle programmatique, il peut tre ncessaire de faire la distinction entre le contexte du jeu de donnes fonctionnel et le contexte de vue historique. Soit parce que les permissions attendues sont diffrentes, soit parce que certains champs ne sont pas prsents dans la structure de l'historique. C'est le cas en ce qui concerne les champs de jeux de donnes, les valeurs calcules, ainsi que les champs dont l'historique a t dsactiv [p 274]. Les mthodes `Adaptation.isHistory`[API] et `AdaptationTable.getHistory`[API] peuvent alors tre utilises dans la rgle programmatique afin d'implmenter un comportement spcifique pour l'historique.

Vue historique des transactions

La vue historique des transactions permet d'accder aux transactions excutes, indpendamment d'une table, du jeu de donnes ou du modle de donnes, directement partir de l'interface utilisateur.

Pour afficher la table 'Historique des transactions', aller dans 'Administration' et slectionner 'Historique et journaux' l'aide du menu droulant symbolis par une flche dans le panneau de navigation. On peut aussi accder l'historique des transactions partir des 'Espaces de donnes' en slectionnant un espace de donnes historis et en utilisant le menu 'Actions' dans l'espace de travail.

Pour plus d'informations, voir vue historique des transactions [p 29].

45.4 Accs SQL l'historique

Cette section dcrit le moyen d'accder directement l'historique grce SQL.

Voir aussi*Accs SQL aux donnes en mode relationnel [p 271]*

Restrictions d'accs

L'accs aux tables de la base de donnes n'est autoris qu'en mode lecture seule. C'est l'administrateur de la base de donnes d'en interdire l'accs en criture, sauf pour l'utilisateur de base de donnes de TIBCO EBX, tel qu'indiqu dans la section Rules for the database access and user privileges [p 397].

Prsentation du schma relationnel

Voici une description des tables d'historique dans la base de donnes.

Le schma de la base de donnes contient (voir aussi le diagramme dans la section suivante):

Des tables communes et gnriques	La table principale est `HV_TX`;chaque enregistrement de cette table reprsente une transaction. Seules les transactions qui impliquent au moins une table historise sont enregistres. Ces tables communes ont toutes le prfixe "HV".
Tables spcifiques gnres	Pour chaque table historise, une table d'historique spcifique est gnre. Cette table contient l'historique des modifications de donnes sur la table. Dans l'interface utilisateur d'EBX, le nom de cette table en base de donnes peut tre connu en cliquant sur le panneau de documentation de la table (mode avanc). Toutes les tables d'historique spcifiques ont le prfixe "HG".

Exemple de table d'historique gnre

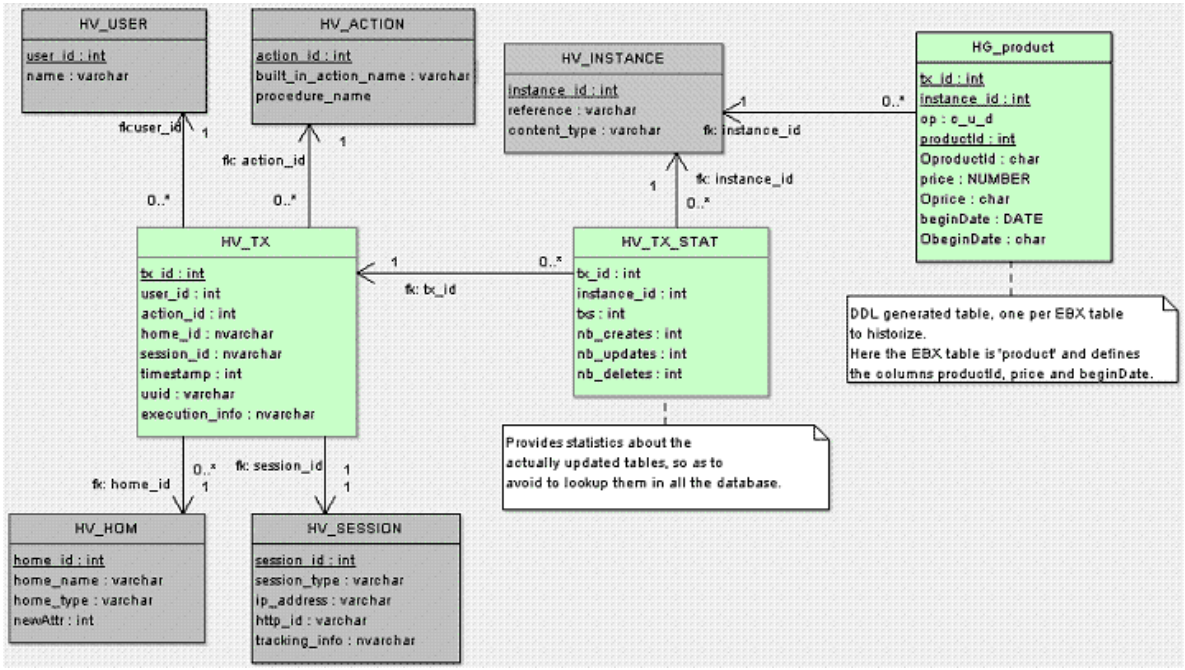
Dans l'exemple suivant, une table nomme `product` est historise. Supposons que cette table dclare trois champs dans le modle de donnes EBX:

Produit

- productId:int
- price:int

- beginDate:Date

Le diagramme ci-dessous montre le schma relationnel qui en rsulte:



L'activation de l'historique sur cette table gnre la table HG_product affiche dans la structure du schma de l'historique ci-dessus. Voici la description de ses diffrents champs:

- `tx_id`: ID de la transaction.
- `instance`: ID de l'instance.
- `op`: type d'opration - C (crer), U (mettre jour) ou D (effacer).
- `productId`: `productId` valeur de champ.
- `OproductId`: champ d'opration pour `productId`, voir section suivante.
- `price`: `price` valeur de champ.
- `Oprice`: champ d'opration pour `price`, voir section suivante.
- `beginDate`: `date` valeur de champ.
- `ObeginDate`: champ d'opration pour `beginDate`, voir section suivante.

Combinaison doperations

Si plusieurs oprations ont lieu au cours de la mme transaction, le champ d'opration est rsolu selon la rgle suivante :

- `C + U -> C`

  `D + U -> D`

  `D + C -> U`

  `C + D -> {} (pas d'enregistrement dans l'historique)`

Valeurs de champs d'opration

Pour chaque champ fonctionnel, un champ d'opration supplmentaire est dfini. Il est compos du nom du champ dont le prfixe est `O`. Ce champ prcise si le champ fonctionnel a t modifi et prend une des valeurs suivantes :

- `null`: si la valeur du champ fonctionnel n'a pas t modifie (et que sa valeur n'est pas INHERIT).
- `M`: si la valeur du champ fonctionnel a t modifie (mais ne prend pas la valeur INHERIT).
- `D`: si l'enregistrement a t effac.

Si inheritance [p 294] est activ, le champ d'opration peut avoir trois valeurs supplmentaires :

- `T`: si la valeur du champ fonctionnel n'a pas t modifie et que sa valeur est INHERIT.
- `I`: si la valeur du champ fonctionnel a pris la valeur INHERIT.
- `O`: si l'enregistrement a t mis en mode OCCULTING.

45.5 Impacts et limitations du mode historis

La fonctionnalit d'historique a certains impacts et des limitations connues, qui sont dcrits dans cette section. Lors de l'utilisation du mode historis, il est fortement conseill de lire attentivement ces limitations et de contacter le support de Cloud Software Group, Inc. pour toute question.

Validation

Certaines contraintes du modle de donnes d'EBX deviennent bloquantes quand l'historique de la table est activ. Pour plus d'informations, voir la section Contraintes structurelles [p 269].

Restrictions du modle de donnes pour les tables historises

Certaines restrictions s'appliquent aux modles de donnes qui contiennent des tables historises :

- Restrictions sur le modle de donnes suite au mode mapp [p 265]
- Certaines limitations existent pour deux types de listes agrges : les listes agrges sous une autre liste agrge, et les listes agrges sous un groupe terminal. Les modles de donnes contenant ce type de listes agrges peuvent tre utiliss, cependant ces listes seront ignores (non historises).
- Les valeurs calcules sont ignores.
- La prsence d'attributs dfinis par l'utilisateur sur des tables historises provoque des erreurs lors de la compilation du modle de donnes.

Les volutions du modle de donnes peuvent aussi tre restreintes par le SGBDR sous-jacent, en fonction des donnes dj prsentes dans les tables concernes.

Voir aussi*volutions du modle de donnes [p 287]*

Autres limitations du mode historis

- Aucune donne n'est copie lorsque l'historique est activ sur une table contenant des donnes existantes.
- Les oprations d'ensemble sur les jeux de donnes ne sont pas historises (crer et supprimer une instance), mme si elles dclarent une table historise.

- Les libells par dfaut rfrenant un champ non historis ne sont pas pris en charge pour les tables historises.

  Par consquent, les libells par dfaut rfrenant un champ calcul ne sont pas pris en charge pour les tables historises.

  Pour contourner ce problme, il suffit d'implmenter l'interface `UILabelRenderer` et d'adapter le calcul du libell pour l'historique.
- D3:l'historique peut tre activ dans l'espace de donnes de destination d'un noeud maître. Mais, dans l'espace de donnes de destination des noeuds esclaves, la fonction historisation est toujours dsactive.
- Utilisateur enregistr dans l'historique:pour certaines oprations spcifiques, l'utilisateur ralisant la dernire opration et celui qui est enregistr dans l'historique correspondant peuvent tre diffrents.

  Ceci est d au fait que ces oprations sont un rapport du statut des donnes un tat antrieur :

  - Import d'archives:lors de l'import d'archive dans un espace de donnes, l'heure et le nom de l'utilisateur de la dernire opration ralise dans l'espace de donnes sont conservs, alors que le nom de l'utilisateur enregistr dans l'historique est celui qui ralise l'import.
  - Fusion programmatique:lors de la fusion programmatique d'un espace de donnes, l'heure et le nom de l'utilisateur de la dernire opration ralise dans l'espace de donnes enfant sont conservs, alors que le nom de l'utilisateur enregistr dans l'historique est celui qui ralise la fusion.
  - D3:en ce qui concerne la fonctionnalit de "distributed data delivery", lorsqu'une diffusion est ralise, les donnes du noeud maître sont reportes dans le noeud esclave. L'heure, ainsi que le nom de l'utilisateur qui a ralis la dernire opration dans l'espace de donnes enfant, sont conservs. Le nom de l'utilisateur enregistr dans l'historique est "ebx-systemUser";c'est lui qui procde la dclaration sur le noeud esclave lors de la diffusion.

CHAPITRE 46

Rplication

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Configuration de la rplication
3. Accder une table rplique l'aide de SQL
4. Requte d'actualisation des rplications 'onDemand'
5. Impact et limitations de la rplication

46.1 Prsentation

Les donnes stockes dans le rfrentiel de TIBCO EBX peuvent tre copies en miroir dans des tables relationnelles ddies afin d'offrir un accs direct aux donnes via des requtes et des vues SQL.

Tout comme l'historique et le mode relationnel, cette rplication des donnes est transparente pour l'utilisateur final et pour les applications clientes. Certaines actions provoquent des modifications automatiques des rpliques dans la base de donnes:

- L'activation de la rplique, au niveau du modle, met jour le schma de la base de donnes en excutant automatiquement les dclarations de DDL ncessaires.
- Les volutions du modle de donnes impactant les tables rpliques, telles que la cration d'une nouvelle colonne, actualisent aussi le schma de la base de donnes par le biais de dclarations DDL.
- En utilisant le mode d'actualisation 'onCommit':la mise jour des donnes du rfrentiel d'EBX dclenche les insertions, mises jour et suppressions associes aux rpliques en base de donnes.

Voir aussi

Mode relationnel *[p 267]*

Historique *[p 273]*

volutions du modle de donnes *[p 287]*

Repository administration *[p 396]*

> **Note**
>
> *table rplique*:dsigne une table de donnes de rfrence qui a fait l'objet d'une rplication
>
> *table de rplique* (ou *rplique*):dsigne une table en base de donnes qui est la cible de la rplication

46.2 Configuration de la rplication

Activer la rplication

Pour dfinir une *unit de rplication* sur un modle de donnes, utiliser l'lment `osd:replication` sous les lments `annotation/appinfo`. Chaque unit de rplication dsigne des tables dans un jeu de donnes unique dans un espace de donnes spcifique.

Les lments imbriqus sont les suivants :

lment	Description	Requis
`name`	Nom de l'unit de rplication. Ce nom identifie l'unit de rplication dans le modle de donnes. Ce nom doit tre unique.	Oui
`dataSpace`	Indique l'espace de donnes concern par la rplication. Cet espace de donnes ne peut ni tre une version ni tre relationnel.	Oui
`dataSet`	Indique le jeu de donnes concern par la rplication.	Oui
`refresh`	Spcifie la politique de synchronisation des donnes. Les politiques disponibles sont : • `onCommit`: Dans la base de donnes, le contenu de la rplique est toujours jour par rapport sa table source. Chaque transaction qui met jour la table source d'EBX dclenche les insertions, mises jour et suppressions correspondantes sur la rplique. • `onDemand`:La rplication des tables spcifies n'est effectue que lorsqu'une opration d'actualisation explicite est ralise. Voir Requte d'actualisation des rplications 'onDemand' [p 284].	Oui
`table/path`	Indique le chemin de la table dans le modle de donnes qui doit tre rplique dans la base de donnes.	Oui
`table/nameInDatabase`	Indique le nom de la table dans la base de donnes qui contiendra les donnes rpliques. Ce nom doit tre unique par rapport toutes les units de rplication.	Oui
`table/element/path`	Indique le chemin de la liste agrge dans la table qui doit tre rplique dans la base de donnes.	Oui
`table/element/ nameInDatabase`	Indique le nom de la table dans la base de donnes qui contiendra les donnes rpliques de la liste agrge. Ce nom doit tre unique par rapport toutes les units de rplication.	Oui

Par exemple :

```
<xs:schema>
 <xs:annotation>
  <xs:appinfo>
   <osd:replication>
    <name>ProductRef</name>
    <dataSpace>ProductReference</dataSpace>
    <dataSet>productCatalog</dataSet>
    <refresh>onCommit</refresh>
    <table>
     <path>/root/domain1/tableA</path>
     <nameInDatabase>PRODUCT_REF_A</nameInDatabase>
```

```
    </table>
    <table>
     <path>/root/domain1/tableB</path>
     <nameInDatabase>PRODUCT_REF_B</nameInDatabase>
     <element>
      <path>/retailers</path>
      <nameInDatabase>PRODUCT_REF_B_RETAILERS</nameInDatabase>
     </element>
    </table>
   </osd:replication>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:schema>
```

Remarques :

- Voir Restrictions du modle de donnes pour les tables rpliques [p 284]
- Si, lors de la compilation du modle de donnes, le jeu de donnes et/ou l'espace de donnes spcifis n'existent pas dans le rfrentiel courant, un avertissement est mis, mais la table des rplicats est cre dans la base de donnes. Une fois que l'espace de donnes et le jeu de donnes sont crs, la rplication est active.
- Si, lors de la compilation d'un modle de donnes, une table rplique est efface, ou si certaines des proprits ci-dessus ont t modifies, la rplique est efface de la base de donnes, puis recre avec la nouvelle dfinition si besoin.

Dsactiver la rplication pour un champ ou un groupe donn

Le comportement par dfaut d'une table rplique est la copie de tous ses lments pris en charge (voir Restrictions du modle de donnes pour les tables rpliques [p 284]).

Il est possible de dsactiver la rplication pour un champ ou un groupe spcifique, soit grce l'assistant de modle de donnes, soit en modifiant le modle de donnes sous-jacent.

Pour dsactiver la rplication d'un champ ou d'un groupe en modifiant le modle de donnes, utiliser l'lment `osd:replication` avec l'attribut `disable="true"`.

```
<xs:element name="longDescription" type="xs:string">
    <xs:annotation>
        <xs:appinfo>
            <osd:replication disable="true" />
        </xs:appinfo>
    </xs:annotation>
</xs:element>
```

Pour dsactiver la rplication d'un champ ou d'un groupe l'aide de l'assistant du modle de donnes, utiliser la proprit `Replication` des `Advanced properties` de l'lment.

Lorsque cette proprit est dfinie sur un groupe, la rplication est dsactive rcursivement pour tous ses descendants. Une fois que la rplication a t dsactive sur un groupe, il n'est pas possible de la r-activer sur un descendant.

> **Note**
>
> Si la table contenant le champ ou le groupe n'est pas rplique, cette proprit n'aura aucun effet.
>
> Il n'est pas possible de dsactiver la rplication pour les champs de cl primaire.

46.3 **Accder une table rplique l'aide de SQL**

Cette section dcrit comment accder directement une rplique en utilisant SQL.

Voir aussi*Accs SQL l'historique [p 276]*

Trouver la rplique dans la base de donnes

Pour chaque table de rplique d'EBX, une table correspondante est gnre dans le SGBDR. Grce l'interface utilisateur d'EBX, il est possible de trouver le nom de cette table de base de donnes en cliquant sur le panneau de documentation de la table.

Restrictions d'accs

L'accs direct aux rpliques en base de donnes n'est autoris qu'en mode lecture-seule. Il est de la responsabilit de l'administrateur de la base de donnes de bloquer l'accs en criture tous les utilisateurs de la base de donnes l'exception de celui utilis par EBX.

Voir aussi*Rules for the database access and user privileges* *[p 397]*

Lectures SQL

Les lectures directes en SQL sont possibles sur des transactions bien encadres et de prfrence de courte dure. Toutefois, pour de tels accs, les permissions d'EBX ne sont pas prises en compte. Par consquent, les applications ayant le droit d'effectuer des lectures doivent tre scurises par le biais de permissions et de processus d'authentification tiers.

46.4 Requte d'actualisation des rplications 'onDemand'

La politique d'actualisation 'onDemand' ncessite une requte explicite pour actualiser les donnes de la table rplique.

Il y a plusieurs moyens pour demander une actualisation de rplication :

- **Interface utilisateur**:Dans le menu d'actions du jeu de donnes, utiliser l'action 'Rafraîchir les rpliques' sous le groupe 'Rplication' pour lancer l'assistant d'actualisation de rplication.
- **Services de donnes**: Utiliser l'opration de services de donnes de l'actualisation des rplications. Voir Replication refresh [p 666] pour davantage d'informations concernant les services de donnes.
- **API Java**:Appeler les mthodes `ReplicationUnit.performRefresh`[API] dans l'API `ReplicationUnit` afin d'actualiser l'unit de rplication.

46.5 Impact et limitations de la rplication

La fonctionnalit de rplication a certains impacts et des limitations connues, lists ci-dessous. En cas d'utilisation de la rplication, il est fortement conseill de lire attentivement ces limitations et de contacter le support de Cloud Software Group, Inc. pour toute question.

Voir Supported databases [p 337] pour les bases de donnes qui prennent en charge la rplication.

Validation

Certaines contraintes de modle de donnes d'EBX deviennent bloquantes lorsque la rplication est active. Pour plus d'informations, voir Contraintes structurelles [p 269].

Restrictions du modle de donnes pour les tables rpliques

Certaines restrictions s'appliquent aux modles de donnes contenant des tables rpliques :

- Restrictions sur le modle de donnes suite au mode mapp [p 265]

- L'hritage de jeu de donnes n'est pas pris en charge dans le cas de la politique d'actualisation 'onCommit' si le jeu de donnes spcifi n'est pas un jeu de donnes racine ou s'il n'a pas encore t cr. Voir hritage de jeu de donnes [p 295] pour plus d'informations.
- De la mme manire, l'hritage de champ est uniquement pris en charge pour la politique d'actualisation 'onDemand'. Cela signifie qu' la compilation du modle de donnes, une erreur est mise si le mode d'actualisation est 'onCommit' et que la table devant tre rplique a un champ hrit. Voir champs hrits [p 296] pour plus d'informations.
- Les valeurs calcules sont ignores.
- Il existe des limitations pour deux catgories de listes agrges : les listes agrges sous une autre liste agrge, et les listes agrges sous un groupe terminal. Les modles de donnes contenant de telles listes agrges peuvent tre utilises, toutefois ces listes seront ignores (non repliques).
- Les attributs personnaliss ne sont pas pris en charge. Une erreur de compilation est mise s'ils sont inclus dans une unit de rplication.

Les volutions du modle de donnes peuvent aussi tre limites par le SGBDR sous-jacent, en fonction des donnes dj contenues dans les tables concernes.

Voir aussi*volutions du modle de donnes [p 287]*

Configuration de la base de donnes

L'opration d'actualisation est optimise pour transmettre uniquement les lignes de la table source qui ont t modifies (cration et suppression) depuis la dernire actualisation. Toutefois, en fonction du volume de donnes changes, cette opration peut se rvler volumineuse et ncessiter des transactions importantes. En particulier, la premire opration d'actualisation peut concerner un grand nombre de lignes. Il est ncessaire que la base de donnes soit configure correctement afin de permettre l'excution de ces transactions de manire optimale.

Par exemple, avec Oracle :

- Il est impratif que l'ensemble des tables de rplique dans une unit de rplication puisse tre contenu dans l'espace de table 'UNDO'.
- Il est recommand de fournir suffisamment d'espace dans le cache de donnes afin de permettre ces transactions de s'excuter avec un minimum d'accs disque.
- Il est recommand de rserver suffisamment d'espace pour les groupes de logs 'REDO' pour que les transactions n'aient pas attendre le processus 'db_writer'.

Distributed data delivery (D3)

La rplication est disponible sur le master D3 ainsi que sur les espaces de donnes de livraison esclaves. Sur le master, le comportement de rplication est le mme que dans un espace de donnes smantique standard, mais sur les esclaves, le contenu rpliqu est celui de la dernire image diffuse.

Dans un espace de donnes de livraison esclave, certaines restrictions s'appliquent :

- La politique d'actualisation dfinie dans le modle de donnes n'a aucune influence sur le comportement dcrit plus haut:la rplication se dclenche toujours la suite de la cration de l'image.
- L'action `Refresh replicas` n'est pas disponible.
- L'invocation de la mthode `ReplicationUnit.performRefresh`[API] n'est pas autorise.

Voir aussi*Présentation D3* *[p 448]*

Autres limitations de la réplication

- Limitations des bases de données prises en charge [p 337]
- Dans le cas de l'héritage, un champ d'enregistrement répliqué ne peut être une "valeur héritée" (`AdaptationValue.INHERIT_VALUE`). Il ne fait que contenir la valeur héritée dans ce genre de cas. Plus généralement, il n'est pas possible de distinguer l'état héritage de l'état écriture.

CHAPITRE 47

volutions du modle de donnes

Ce chapitre prsente les modifications qui peuvent tre ralises sur les modles de donnes, ainsi que les ventuelles limitations. Les restrictions et/ou impacts potentiels des volutions du modle de donnes dpendent du mode de persistance. Les principes applicables chaque mode sont les suivants :

- Mode smantique:flexible et non-bloquant. Peut conduire une perte de donnes, par exemple une dfinition de cl primaire peut voluer librement, mais tout enregistrement existant, dans un espace de donnes ou dans un snapshot, qui enfreint la contrainte de cl primaire ne sera plus charg.
- Tout mode mapp:restrictif et donc bloquant si les donnes existent et si l'volution viendrait enfreindre leur intgrit.

Attention

A chaque volution du modle de donnes, il est crucial d'anticiper la perte potentielle de donnes. Si la perte est avre, dans le cas o les donnes existantes doivent tre conserves, un plan de migration de donnes doit tre mis en place et appliqu avant la publication ou le dploiement du nouveau modle de donnes. Il est important de noter galement que les donnes ne sont pas dtruites immdiatement aprs l'volution du modle de donnes;en mode smantique, du moment qu'aucune mise jour n'est effectue sur une table dont la dfinition a volu, si le modle de donnes est publi un tat prcdent, alors les donnes prcdentes sont rcupres.

> **Note**
>
> Certains types d'volutions du modle de donnes ne peuvent tre raliss directement dans l'interface utilisateur. Le modle de donnes doit alors tre export, modifi au format XSD, puis r-import. En ce qui concerne les modifications d'un modle de donnes impactant galement sa configuration, en plus de sa structure, le XSD doit tre import dans TIBCO EBX partir d'un module. Sans quoi, les modifications de configuration ne seront pas prises en compte.

Voir aussi*Mode mapp* *[p 265]*

Ce chapitre contient les sections suivantes :

1. Types d'volutions autoriss
2. Limitations/restrictions

47.1 Types d'volutions autoriss

Cette section dcrit les modifications possibles sur les modles de donnes aprs leur cration.

volutions au niveau du modle

Les modifications suivantes peuvent tre effectues sur les modles de donnes existants :

- Un modle de donnes, en mode smantique, peut tre dclar comme tant en mode relationnel. Les donnes doivent tre transfres manuellement, en exportant puis r-important un fichier XML ou une archive.
- Le mode relationnel peut tre dsactiv sur le modle de donnes. Les donnes doivent tre transfres manuellement, en exportant puis r-important un fichier XML ou une archive.
- Des units de rplication peuvent tre ajoutes au modle de donnes. Si leur politique d'actualisation est 'onCommit', les rpliques correspondantes seront cres et actualises lors de la prochaine compilation de schma.
- Les units de rplication peuvent tre retires du modle de donnes. Les rpliques correspondantes seront immdiatement abandonnes.
- Le modle de donnes peut tre supprim. S'il dclare des units de rplication, les rpliques correspondantes seront immdiatement abandonnes. S'il est relationnel ou contient des tables historises, cette modification marque les tables mappes associes comme dsactives. Voir Mapping en bases de donnes [p 429] pour la suppression effective des objets de base de donnes associs.

volutions au niveau de la table

Le modle de donnes peut subir les modifications suivantes au niveau de la table :

- Une nouvelle table peut tre ajoute. Lors de la cration, la table peut aussi dclarer un ou plusieurs modes mapps.
- Une table existante peut tre supprime. Si elle dclare des units de rplication, les rpliques correspondantes seront immdiatement abandonnes. Cette modification marque la table mappe comme dsactive, si elle est historise ou relationnelle. Voir Mapping en bases de donnes [p 429] pour la suppression effective des objets de base de donnes associs.
- Une table existante en mode smantique peut tre dclare comme tant en mode relationnel. Les donnes doivent tre transfres manuellement, en exportant puis r-important un fichier XML ou une archive.
- L'historique peut tre activ ou dsactiv sur une table. Il ne prendra pas en compte les oprations ralises pendant sa priode de dsactivation.
- Une table peut tre renomme. Les donnes doivent tre transfres manuellement, en exportant puis r-important un fichier XML ou une archive, car cette modification est considre comme tant une combinaison de suppression et de cration.

volutions au niveau du champ

Le modle de donnes peut subir les modifications suivantes au niveau du champ :

- Un nouveau champ peut tre ajout.
- Un champ existant peut tre supprim. En mode smantique, les donnes du champ supprim seront retires de chaque enregistrement lors de sa prochaine mise jour. Pour une table de rplique, la

colonne correspondante est automatiquement supprime. En mode historique ou relationnel, le champ est marqu comme dsactiv.

- Un champ peut tre spcifiquement dsactiv de l'historique ou de la rplication qui s'appliquent sa table, en utilisant l'attribut `disable="true"`. Pour une table rplique, la colonne correspondante est automatiquement retire. Pour une table d'historique, la colonne reste prsente mais est marque comme dsactive. Voir Dsactivation de l'historique d'un champ ou d'un groupe spcifique [p 274] et Dsactiver la rplication pour un champ ou un groupe donn [p 283].
- Les facettes d'un champ peuvent tre modifies, except celles listes sous Limitations/restrictions [p 289].

Les modifications nonces ci-dessus sont acceptes, mais peuvent donner lieu une perte de donnes. Les donnes doivent tre transfres manuellement, en exportant puis en r-important un fichier XML ou une archive, car ces modifications sont considres comme tant la combinaison d'une suppression et d'une cration.

- Un champ peut tre renomm.
- Le type d'un champ peut tre chang.

volutions au niveau de l'index

- Un index peut tre ajout ou renomm.
- Un index peut tre modifi, en modifiant ses champs ou en changeant leur ordre. En mode mapp, l'index existant est effac et un autre est cr.
- Un index peut tre supprim. En mode mapp, un index supprim est aussi supprim de la base de donnes.

47.2 Limitations/restrictions

> **Note**
>
> Toutes les limitations listes dans cette section, qui impactent le mode mapp, peuvent tre contournes en purgeant les ressources de base de donnes de la table mappe. Pour la procdure de purge des ressources de base de donnes de la table mappe, voir Mapping en bases de donnes [p 429].

Limitations lies aux volutions de cl primaire

Lorsqu'une dfinition de cl primaire est modifie :

- En mode smantique, les enregistrements existants sont chargs dans le cache uniquement s'ils :
  - Respectent la contrainte d'unicit de la cl primaire,
  - Sont conformes la nouvelle structure de la cl primaire.
- En mode mapp, le SGBDR sous-jacent accepte uniquement une modification de cl primaire si tous les enregistrements de la table respectent ses contraintes d'unicit et de non-nullit. En particulier, si une table contient dj des enregistrements :
  - L'ajout d'un nouveau champ la cl primaire ncessite l'attribution d'une valeur par dfaut pour ce champ. Contournement : d'abord ajouter le champ, le valoriser pour les enregistrements existants, puis ajouter le champ la cl primaire.

- La suppression d'un champ existant de la cl primaire sera rejete si elle implique de rendre non unique la cl primaire d'enregistrements existants (l'attribution d'une valeur par dfaut ne change rien dans ce cas).
- En rgle gnrale, il n'est pas possible de renommer un champ de cl de primaire;ceci est uniquement possible si ce champ n'tait pas ncessaire pour rendre toutes les cls primaires uniques. En effet, le renommage d'un champ se traduit par une combinaison de suppression et de cration;par consquent, l'opration sera rejete si elle implique de rendre non unique la cl primaire d'enregistrements existants (l'attribution d'une valeur par dfaut ne change rien dans ce cas).

Limitations lies aux volutions de cl trangre

- Lorsque la dclaration d'une facette `osd:tableRef` est ajoute ou modifie, ou que la cl primaire de la table cible d'une facette `osd:tableRef` est modifie :
  - En mode smantique, les valeurs du champ sont uniquement charges dans le cache si elles respectent la nouvelle structure de la cl primaire cible.
  - En mode mapp, la structure d'un champ de cl trangre est dfinie pour correspondre celle de la cl primaire cible. Un champ unique dclarant une contrainte `osd:tableRef` peut alors tre divis en colonnes, dont le nombre et les types correspondent ceux de la cl primaire cible. Par consquent, les cas d'volution suivants auront un impact sur la structure de la table mappe :
    - dclaration d'une nouvelle contrainte `osd:tableRef` sur un champ de table ;
    - suppression d'une contrainte `osd:tableRef` existante sur un champ de table ;
    - ajout (respectivement suppression) d'une colonne (respectivement d') une cl primaire rfrence par une contrainte `osd:tableRef` existante ;
    - modification du type ou du chemin de n'importe quelle colonne d'une cl primaire rfrence par une contrainte `osd:tableRef` existante.

    Ces diffrents cas d'volution se traduisent par une combinaison de crations et/ou de suppressions de champs. Par consquent, les donnes existantes doivent tre transfres manuellement.

Limitations lies aux volutions au niveau du champ

- En mode mapp, lorsqu'une facette `maxLength`, `length`, `totalDigits` ou `fractionDigits` est modifie :

  L'acceptation ou non de cette modification dpend du SGBD sous-jacent, ainsi que du type de champ et du contenu de la table.

  Par exemple, Oracle acceptera de changer une VARCHAR(20) en VARCHAR(50), mais ne changera une VARCHAR(50) en VARCHAR(20) que si la table ne contient pas de valeurs faisant plus de 20 caractres.

  PostgreSQL connaît les mmes limitations. Par ailleurs, toute modification d'un type de champ (incluant les modifications de sa longueur) invalidera toutes les dclarations pr-tablies qui y sont lies et ncessitera un redmarrage du serveur d'application.
- Lorsque la cardinalit d'un lment est modifie :
  - En mode smantique, ce changement est pris en charge. Cependant, deux cas sont distinguer :

  - Lors de la transformation d'un lment unique vers une liste agrge, la valeur unique prcdente est conserve et ajoute la nouvelle liste agrge.
  - Lors de la transformation d'une liste agrge en un lment unique, seule la dernire valeur de la liste agrge est conserve dans l'lment unique. Les autres valeurs sont perdues.
- En mode relationnel, les listes agrges ne sont pas prises en charge. Un message d'erreur est ajout au rapport de compilation du modle de donnes si un lment est chang en liste agrge.
- En mode historis, lorsqu'un lment unique est transform en liste agrge, la modification est prise en compte, mais la valeur unique prcdente est perdue.

Divers

CHAPITRE **48**

Hritage et rsolution de valeur

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Hritage de jeu de donnes
3. Champs hrits
4. Service d'optimisation et refactoring

48.1 Prsentation

Le principe de l'hritage est de mutualiser les ressources qui sont partages par de multiples contextes ou entits. TIBCO EBX propose des mcanismes pour dfinir, factoriser et rsoudre les valeurs de donnes : *hritage de jeu de donnes* et *champs hrits*.

De plus, des *fonctions* peuvent tre dfinies pour calculer des valeurs.

> **Note**
>
> Les mcanismes d'hritage dcrit dans ce chapitre ne doivent pas tre confondus avec l'hritage structurel, lequel s'applique gnralement aux modles et est propos dans des diagrammes de classes UML par exemple.

Voir aussi*Hritage (glossaire)* *[p 27]*

Hritage de jeu de donnes

L'hritage de jeu de donnes est particulirement utile quand les donnes s'appliquent des contextes globaux d'entreprise, tels que filiales ou partenaires.

En se basant sur une hirarchie de jeux de donnes, il est possible de factoriser les donnes communes au niveau de la racine ou des jeux de donnes intermdiaires et de dfinir des donnes spcialises dans des contextes donns.

Les mcanismes d'hritage du jeu de donnes sont dtaills ci-dessous dans Hritage de jeu de donnes [p 295].

Champs hrits

Contrairement l'hritage de jeu de donnes, qui utilise les relations globales entre jeux de donnes, les champs hrits utilisent des dpendances plus fines qui sont spcifiques la structure des donnes. Cela permet de factoriser et de spcialiser les donnes au niveau des entits mtier.

Par exemple, si le modle prcise qu'un "Produit" est associ une "FamilleDeProduits", il est possible que certains attributs du "Produit" hritent leurs valeurs des attributs dfinis dans sa "FamilleDeProduits" associe.

> **Note**
>
> Lors de l'utilisation des deux hritages sur un mme jeu de donnes, l'hritage de champ est prioritaire sur celui du jeu de donnes.

Valeurs calcules (fonctions)

Dans le modle de donnes, il est aussi possible de dfinir qu'un noeud contient une *valeur calcule*. Dans ce cas, la fonction JavaBean spcifie sera excute chaque fois que la valeur est requte.

La fonction peut prendre en compte le contexte courant, tel que les valeurs de l'enregistrement courant ou des calculs utilisant une autre table et envoyer des requtes des systmes tiers.

Voir aussi*Computed values* *[p 551]*

48.2 Hritage de jeu de donnes

Dclaration d'hritage de jeu de donnes

Le mcanisme d'hritage du jeu de donnes est dclar dans un modle de donnes de la faon suivante :

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ebxbnd="urn:ebx-schemas:binding_1.0">
  <xs:annotation>
    <xs:appinfo>
      <osd:inheritance>
        <dataSetInheritance>all</dataSetInheritance>
      </osd:inheritance>
    </xs:appinfo>
  </xs:annotation>
   ...
</xs:schema>
```

L'lment `osd:inheritance` dfinit la proprit `dataSetInheritance` qui permet d'activer ou non l'hritage pour les jeux de donnes utilisant ce modle de donnes. Les valeurs suivantes peuvent tre spcifies :

- `all`, indique que l'hritage est activ pour tous les jeux de donnes qui instancient le modle de donnes.
- `none`, indique que l'hritage est dsactiv pour tous les jeux de donnes qui instancient le modle de donnes.

Si rien n'est spcifi, le mcanisme d'hritage est dsactiv.

Mcanisme de rsolution

Le mcanisme de rsolution des valeurs hrites du jeu de donnes a le comportement suivant :

1. Si la valeur est dfinie localement, elle est retourne.

   Elle peut tre explicitement `null`.

2. Sinon, la premire valeur dfinie localement est recherche en remontant la relation enfant-parent dans la hirarchie des jeux de donnes.

3. Si aucune valeur dfinie n'est trouve, la valeur par dfaut est retourne.

   Si aucune valeur par dfaut n'est dfinie, `null` est retourn.

   Note : Les valeurs par dfaut ne peuvent pas tre dfinies sur :

- Un noeud de cl primaire unique
- Des noeuds auto-incrments
- Des noeuds dfinissant une valeur calcule

Mcanisme de rsolution d'enregistrement

Tout comme les valeurs, les enregistrements d'une table peuvent aussi tre hrits en tant qu'unit par des contextes multiples, mais aussi tre partiellement redfinis (*cras*), dfinis pour un contexte spcifique (*mode racine*), ou tre *occults*.

Un enregistrement se dfinit suivant l'un des quatre modes de dfinition distincts :

enregistrement racine	Dfini localement dans la table et n'a aucun parent. Cela signifie qu'aucun enregistrement ayant la mme cl primaire n'existe dans la table parente, ou que ce parent est un enregistrement occult.
enregistrement surcharg	Dfini localement dans la table et possde un enregistrement parent. Cela signifie qu'un enregistrement ayant la mme cl primaire existe dans la table parente et que ce parent n'est pas un enregistrement occult. Les valeurs de l'enregistrement surcharg sont hrites de son parent, sauf pour les valeurs qu'il redfinit explicitement.
enregistrement hrit	Non dfini localement dans la table courante et possde un enregistrement parent. Toutes les valeurs sont hrites. Les fonctions sont toujours rsolues dans le contexte de l'enregistrement courant et ne sont pas hrites.
enregistrement occult	Prcise que si un parent ayant la mme cl primaire est dfini, ce parent ne sera pas visible dans les descendants de la table.

Voir aussi*Hritage entre jeux de donnes* *[p 143]*

Dfinir le comportement de l'hritage au niveau de la table

Il est aussi possible de dfinir des rgles de gestion dans la dclaration d'une table dans le modle de donnes.

Voir aussi*Properties related to dataset inheritance* *[p 521]*

48.3 **Champs hrits**

Le mcanisme d'hritage spcifique permet de rcuprer la valeur d'un champ selon ses liens avec les autres tables.

Dclaration d'hritage de champ

L'hritage spcifique doit tre dfini sur les noeuds terminaux dans le modle de donnes sous-jacent et est dclar de cette faon :

```
<xs:element name="sampleInheritance" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:inheritance>
    <sourceRecord>
     /root/table1/fkTable2, /root/table2/fkTable3
    </sourceRecord>
    <sourceNode>color</sourceNode>
   </osd:inheritance>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

L'lment `sourceRecord` est une expression qui dcrit comment retrouver l'enregistrement partir duquel la valeur est hrite. Il s'agit d'une cl trangre, ou d'une squence de cls trangres, de l'lment courant la table source.

Si `sourceRecord` n'est pas dfini dans le modle de donnes, les champs hrits sont rcuprs partir de l'enregistrement courant.

L'lment `sourceNode` est le chemin du noeud partir duquel hriter dans l'enregistrement source.

Les conditions suivantes doivent tre remplies pour l'hritage spcifique :

- L'lment `sourceNode` est obligatoire.
- L'expression du chemin de l'enregistrement source doit tre un chemin de cls trangres cohrent, partir de l'lment courant vers l'enregistrement source. Cette expression doit impliquer uniquement des relations un--un et zro--un.
- Le `sourceRecord` ne peut pas contenir d'lments de liste agrge.
- Chaque lment du `sourceRecord` doit tre une cl trangre.
- Si le champ hrit est aussi une cl trangre, le `sourceRecord` ne peut pas faire rfrence lui mme pour obtenir le chemin vers l'enregistrement source de la valeur hrite.
- Chaque lment du `sourceRecord` doit exister.
- Le noeud source doit appartenir la table contenant l'enregistrement source.
- Le noeud source doit tre terminal.
- Le noeud source doit tre accessible en criture.
- Le type de noeud source doit tre compatible avec le type de noeud courant.
- Les cardinalits du noeud source doivent tre compatibles avec celles du noeud courant.
- Le noeud source ne peut tre le mme que celui du champ hrit si les champs desquels on hrite sont rcuprs dans le mme enregistrement.

Mcanisme de rsolution de valeur

Le mcanisme de rsolution pour les valeurs de champs hrits a le comportement suivant :

1. Si la valeur est dfinie localement, elle est retourne.

   Elle peut tre explicitement `null`.

2. Autrement, rsolution de l'enregistrement source ainsi que la valeur hriter, en fonction des proprits dfinies dans le modle de donnes.

3. Le processus est rcursif;si le noeud source ne dfinit pas de valeur localement, elle est alors rsolue en accord avec le comportement d'hritage du noeud source.

48.4 Service d'optimisation et refactoring

EBX offre un service IU intgr pour optimiser l'hritage des jeux de donnes dans la hirarchie des jeux de donnes. Ce service permet les fonctions suivantes :

- **Gestion des valeurs en doublon :** Dtecte et supprime toutes les valeurs qui sont des doublons de la valeur hrite.
- **Mutualisation des valeurs communes :** Dtecte et mutualise les valeurs communes parmi les descendants du mme anctre.

Dtails de la procdure

Les jeux de donnes sont traits de bas en haut, ce qui signifie que si le service est excut sur le jeu de donnes un niveau *N*, avec *N+1* tant le niveau de ses enfants et *N+2* tant le niveau des petits-enfants, le service traitera d'abord les jeux de donnes au niveau *N+2* afin de dterminer s'ils peuvent tre optimiss par rapport aux jeux de donnes du niveau *N+1*. Ensuite, une optimisation du niveau *N+1* sera excute par rapport au niveau *N*.

> **Note**
>
> - Ces fonctions d'optimisation et de refactoring ne grent pas les valeurs par dfaut qui sont dclares dans le modle de donnes.
> - Le niveau le plus lev pris en compte pendant la procdure d'optimisation est toujours le jeu de donnes sur lequel s'excute le service. Cela signifie que l'optimisation et le refactoring ne sont pas raliss entre le jeu de donnes cible et ses propres anctres.
> - L'optimisation de table est ralise sur des enregistrements ayant la mme cl primaire.
> - Les champs hrits ne sont pas optimiss.
> - *Les fonctions d'optimisation et de refactoring ne modifient pas la vue rsolue d'un jeu de donnes, si celle-ci est active.*

Disponibilit du service

Le service 'Optimize & Refactor' est disponible sur les jeux de donnes qui ont des jeux de donnes enfants et dont la proprit 'Active' est rgle sur 'Non' dans les informations du jeu de donnes.

Le service est disponible pour tout profil ayant des droits en criture sur les valeurs du jeu de donnes courant. Il peut tre dsactiv en limitant les droits d'accs d'un profil.

> **Note**
>
> Pour des raisons de performance, les droits d'accs ne sont pas vrifis pour chaque noeud et enregistrement de table.

CHAPITRE 49

Permissions

Les permissions dfinissent l'accs aux donnes et les actions disponibles pour chaque utilisateur.

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Dfinition des rgles utilisateur
3. Dfinition des rgles programmatiques
4. Rsolution des permissions sur donnes
5. Rsolution de permissions pour les services
6. Rsolution de permissions pour les actions

49.1 Prsentation

Les permissions permettent de dfinir si une action est autorise. Elles dfinissent aussi les droits d'accs d'une entit:cache, en lecture, ou en lecture-criture. Les entits principales contrles par les permissions sont les suivantes :

- Espace de donnes
- Jeu de donnes
- Table
- Groupe
- Champ

Utilisateurs, fonctions et profils

La dfinition et la rsolution des permissions sont bases sur la notion de *profils*, terme gnrique faisant rfrence aux utilisateurs ou aux rles.

Chaque utilisateur peut avoir plusieurs rles, et un mme rle peut tre partag par plusieurs utilisateurs.

Ces relations sont dfinies dans le rfrentiel des rles et utilisateurs. Voir Users and roles directory [p 423].

Dfinitions particulires:

- Un *administrateur* est un membre du rle built-in 'ADMINISTRATEUR'.
- Un *propritaire d'un jeu de donnes* est un membre de l'attribut *propritaire* spcifi dans les informations d'un jeu de donnes racine. Dans ce cas, le rle built-in 'PROPRITAIRE' est activ lorsque les permissions sont rsolues dans le contexte du jeu de donnes.

- Un *propritaire d'un espace de donnes* est un membre de l'attribut *propritaire* spcifi pour un espace de donnes. Dans ce cas, le rle built-in 'PROPRITAIRE' est activ lorsque les permissions sont rsolues dans le contexte de l'espace de donnes.

Rgles de permission

Une rgle de permission dfinit les autorisations accordes un profil pour une entit donne.

Les rgles de permission dfinies par l'utilisateur sont cres via l'interface utilisateur. Voir la section Dfinition des rgles utilisateur [p 302].

Les rgles de permission programmatiques peuvent tre cres par des dveloppeurs. Voir la section Dfinition des rgles programmatiques [p 306].

Rsolution des permissions

Les permissions sont toujours rsolues dans le contexte d'une session utilisateur authentifie. Ainsi, les permissions sont principalement bases sur les profils de l'utilisateur.

En gnral, la rsolution est restrictive entre un niveau donn et son niveau parent. Ainsi, pour n'importe quel niveau donn, un utilisateur ne peut pas avoir de permission suprieure celle rsolue au niveau d'un de ses parents.

Les permissions programmatiques sont toujours considres comme tant restrictives.

> **Note**
>
> Dans l'API Java, la classe `SessionPermissions`[API] permet d'accder aux permissions rsolues.

Voir aussi

Rsolution des permissions sur donnes [p 307]

Rsolution de permissions pour les services [p 311]

Rsolution de permissions pour les actions [p 313]

Permissions spciales du propritaire et de l'administrateur

Sur un jeu de donnes

Un administrateur ou un propritaire de jeu de donnes peut effectuer les actions suivantes sur le jeu de donnes :

- Grer ses permissions
- Changer son propritaire, s'il est un jeu de donnes racine
- Modifier ses informations gnrales (libells et descriptions localiss)

> **Attention**
>
> Bien que la dfinition des permissions puisse restreindre les droits d'affichage ou d'excution de certaines actions pour l'administrateur ou le propritaire du jeu de donnes, il reste possible pour ces derniers de modifier leur propre accs, puisqu'ils auront toujours accs la gestion des permissions.

Sur un espace de donnes

Pour devenir le *super propritaire* d'un espace de donnes, un utilisateur doit :

- tre propritaire de l'espace de donnes et tre autoris grer ses permissions, ou
- tre propritaire d'un espace de donnes qui est l'anctre de l'espace de donnes courant et tre autoris grer les permissions de cet espace de donnes anctre.

L'administrateur ou le super propritaire d'un espace de donnes peut raliser les actions suivantes sur l'espace de donnes :

- Grer ses permissions
- Changer son propritaire
- Le verrouiller ou le dverrouiller
- Modifier ses informations gnrales (libells et descriptions localiss)

De plus, dans un workflow, lors de l'utilisation des scripts task intgrs 'Crer un espace de donnes' ou 'Crer une image', les permissions rsolues sont calcules en utilisant le propritaire dfini dans la configuration du script task plutt que dans la session courante. Cela s'explique par le fait que, pour ces cas, la session courante est associe un utilisateur systme.

Attention

Bien que la dfinition des permissions puisse restreindre les droits d'affichage ou d'excution de certaines actions pour l'administrateur ou le propritaire de l'espace de donnes, il reste possible pour ces derniers de modifier leur propre accs, puisqu'ils auront toujours accs la gestion des permissions.

Impact de la fusion sur les permissions

Lorsqu'un espace de donnes est fusionn, les permissions du jeu de donnes de l'espace de donnes enfant sont fusionnes avec celles du jeu de donnes de l'espace de donnes parent si et seulement si l'utilisateur le prcise pendant le processus de fusion. Les permissions de l'espace de donnes parent ne sont jamais impactes.

Si certains lments sont masqus pour le profil qui tente de raliser une fusion, l'action ne sera pas ralisable, car l'impact de la fusion sur les donnes ne sera pas entirement visible.

Aspects importants concernant les permissions

Il faut garder l'esprit les recommandations suivantes lorsqu'on travaille avec les permissions :

- Lorsqu'une permission `hidden` est retourne sur un champ pour une session donne, un utilisateur pourrait "deviner" des informations normalement interdites en filtrant ou en triant sur ce champ dans les cas suivants :
  - En utilisant une `Request`[API] sur cette table avec **les permissions actives** `Request.setSession`[API]. Pour viter cela, les permissions sur ce noeud doivent tre vrifies explicitement avant d'appliquer le filtre ou le critre de tri.
  - En dfinissant une vue personnalise sur cette table dans l'IHM. Pour viter cela, la permission de dfinir des vues personnalises devrait tre restreinte pour ces utilisateurs.
- La rsolution des libells personnaliss de tables (proprit 'defaultlabel') et de relations (proprit 'display') ne tient pas compte des permissions. Les champs habituellement masqus du fait des restrictions de droits d'accs, seront affichs dans ces libells. Aucun de ces libells ne devrait donc

contenir de champ confidentiel. Sinon, une stratgie de permission doit tre dfinie afin de restreindre l'affichage de l'intgralit du libell.

- Lorsqu'une procdure dsactive toutes les vrifications de permission en utilisant `ProcedureContext.setAllPrivileges`[API], le code client doit vrifier que la session de l'utilisateur courant est autorise excuter la procdure.
- Lors de l'excution d'une action sur table (cration, suppression, surcharge, occultation) dans une procdure, le droit d'accs du noeud de la table pour la session utilisateur est ignor durant la rsolution des permissions. Lorsque ce contrle doit tre effectu, le code client doit au pralable appeler explicitement `SessionPermissions.getNodeAccessPermission`[API] dans la procdure.
- Afin d'optimiser la rsolution des permissions, un cache ddi est mis en place au niveau de la session;il ne prend en compte que les rgles dfinies par l'utilisateur, et non les rgles programmatiques (celles-ci ne sont pas mises en cache car elles sont contextuelles et dynamiques). Le cycle de vie du cache de session dpend du contexte, voir l'explication ci-dessous :
  - Dans l'IU, le cache est vid pour chaque vnement non-ajax (i.e affichage de page, ouverte de pop-up, etc.).
  - Dans les procdures programmatiques, le cache n'est vid qu' la fin de la procdure, moins d'tre explicitement vid (voir ci-dessous).

> **Attention**
>
> Lors de la modification de permissions dans un contexte de procdure (import d'une archive EBX ou fusion programmatique d'un espace de donnes), le cache de la session **doit** tre vid via un appel `Session.clearCache`[API]. Autrement, ces modifications ne seront pas prises en compte avant la fin de la procdure.

49.2 Dfinition des rgles utilisateur

Tous les niveaux utilisent un schma similaire, qui permet la dfinition de rgles de permission pour les profils.

Dfinition des rgles utilisateur dans les espaces de donnes

Pour un espace de donnes, les permissions administrables pour chaque profil sont les suivantes :

Mode	Autorisation
criture	• Peut voir l'espace de donnes. • Peut accder aux jeux de donnes en fonction des permissions du jeu de donnes.
Lecture seulement	• Peut voir l'espace de donnes et ses images. • Peut voir les espaces de donnes enfants, si autoris par les permissions. • Peut voir le contenu de l'espace de donnes, sans pouvoir le modifier.
Masqu	• Ne peut voir ni l'espace de donnes ni ses images. • Si autoris voir l'espace de donnes enfant, peut voir l'espace de donnes courant sans pouvoir le slectionner. • Ne peut accder au contenu de l'espace de donnes, ni aux jeux de donnes. • Ne peut raliser aucune action sur l'espace de donnes.

Restriction d'accs	Indique si cette association profil-permission de l'espace de donnes doit tre prioritaire sur les autres rgles de permissions.
Crer un espace de donnes enfant	Indique si le profil peut crer des espaces de donnes enfants partir de l'espace de donnes courant.
Crer une image enfant	Indique si le profil peut crer une image de l'espace de donnes courant.
Initialiser la fusion	Indique si le profil peut fusionner l'espace de donnes courant avec son espace de donnes parent.
Exporter une archive	Indique si le profil peut exporter l'espace de donnes courant comme archive.
Importer une archive	Indique si le profil peut importer une archive dans l'espace de donnes courant.
Fermer un espace de donnes	Indique si le profil peut fermer l'espace de donnes courant.
Fermer une image	Indique si le profil peut fermer une image de l'espace de donnes courant.
Droits sur les services	Indique si un profil a le droit d'excuter des services sur l'espace de donnes. Par dfaut, tous les services d'espaces de donnes sont autoriss. Un administrateur ou super propritaire

	de l'espace de donnes courant ou un utilisateur donn autoris modifier les permissions sur l'espace de donnes courant peut modifier ces permissions afin de restreindre les services de l'espace de donnes pour certains profils.
Permissions de l'espace de donnes enfant la cration	Lorsqu'un utilisateur cre un espace de donnes enfant, les permissions de ce nouvel espace de donnes sont automatiquement attribues au propritaire du profil, conformment aux permissions dfinies sous 'Permissions d'un espace de donnes enfant la cration' dans l'espace de donnes parent. Si plusieurs permissions sont dfinies pour le propritaire via diffrents rles, le profil du propritaire se comporte de la mme manire qu'un autre profil et les permissions sont rsolues [p 300] de la mme manire.

Dfinition des rgles utilisateur dans les jeux de donnes

Pour un jeu de donnes, les permissions administrables pour chaque profil sont les suivantes :

Actions sur les jeux de donnes

Restriction d'accs	Indique si l'association profil-permission du jeu de donnes doit tre prioritaire sur les autres rgles de permissions.
Crer un jeu de donnes enfant	Indique si le profil a le droit de crer un jeu de donnes enfant partir du jeu de donnes courant.
Dupliquer un jeu de donnes	Indique si le profil a le droit de dupliquer le jeu de donnes courant.
Modifier le jeu de donnes parent	Indique si le profil a le droit de changer le jeu de donnes parent d'un jeu de donnes enfant.

Actions sur les tables

Les droits d'actions sur les tables par dfaut sont dfinis au niveau du jeu de donnes. Il est ensuite possible de surcharger ces droits par dfaut pour une ou plusieurs tables donnes. Les permissions administrables pour chaque profil sont les suivantes :

Crer un enregistrement	Indique si le profil a le droit de crer des enregistrements dans la table.
Surcharger un enregistrement	Indique si le profil a le droit de surcharger les enregistrements de la table.
Occulter un enregistrement	Indique si le profil a le droit d'occulter les enregistrements de la table.
Supprimer un enregistrement	Indique si le profil a le droit de supprimer des enregistrements dans la table.

Droits d'accs sur les valeurs de noeud

Les permissions dfinies sur des noeuds terminaux spcifiques surchargent les droits d'accs par dfaut.

Lecture-criture	Peut afficher et modifier les valeurs du noeud.
Lecture	Peut afficher les noeuds, mais ne peut pas modifier leurs valeurs.
Masqu	Ne peut pas voir les noeuds.

Permissions sur les services

Un administrateur ou propritaire de l'espace de donnes courant peut modifier les permissions par dfaut d'un service afin de permettre ou limiter son usage pour certains profils.

Activ	Rend le service disponible pour le profil.
Dsactiv	Rend le service indisponible pour le profil. Il ne sera ni affich dans les menus, ni excutable depuis un composant web.
Par dfaut	Rend le service activ ou dsactiv selon la permission par dfaut dfinie lors de sa dclaration. Voir `ActivationContext.setDefaultPermission`[API] pour plus d'informations.

49.3 Dfinition des rgles programmatiques

Les rgles programmatiques permettent de dfinir plus prcisment les conditions d'accs une donne ou un service utilisateur en fonction du contexte.

Il existe plusieurs types de rgles programmatiques :

- les `AccessRule`[API], dcrites ci-aprs dans la section Dfinition des rgles d'accs aux donnes [p 306].
- les ServiceActivationRule [p 306], dcrites ci-aprs dans la section Dfinition des rgles d'activation sur service [p 306].
- les `ServicePermissionRule`[API], dcrites ci-aprs dans la section Dfinition des rgles de permission sur service [p 307].

Dfinition des rgles d'accs aux donnes

Les `AccessRules` sont des rgles permettant de dfinir programmatiquement et en fonction du contexte les droits de lecture/criture sur un noeud du modle de donnes ou sur les enregistrements d'une table.

La dfinition d'une `AccessRule` s'effectue ainsi :

1. Cration d'une rgle sous la forme d'une classe Java implmentant l'interface `AccessRule`[API] ou `AccessRuleForCreate`[API].
2. Assignation de cette rgle aux noeuds concerns dans l'extension du schma:`SchemaExtensions`[API].

   Selon la cible (noeud(s) du modle ou enregistrements) ou le type (`AccessRule` oi `AccessRuleForCreate`) de la rgle,plusieurs mthodes comme `SchemaExtensionsContext.setAccessRuleForCreateOnNode`[API] ou `SchemaExtensionsContext.setAccessRuleOnOccurrence`[API] sont disponibles.

   La rgle ainsi assigne est dite "locale" et n'est excute que lorsque l'entit cible est sollicite. Voir Rsolution des permissions sur donnes [p 307] pour plus d'informations.

> **Attention**
>
> Une seule `AccessRule` peut tre dfinie pour chaque noeud, espace de donnes ou enregistrement. Une seule `AccessRuleForCreate` peut tre dfinie pour chaque noeud d'une table. Ainsi, la dfinition d'une nouvelle rgle programmatique d'un certain type remplacera l'existante.

Dfinition des rgles d'activation sur service

Les `ServiceActivationRules` sont des rgles permettant de dfinir si un service est activ ou non pour un espace de donnes ou un jeu de donnes. Un service dsactiv par ce moyen n'est jamais disponible dans l'entit pour laquelle il est dsactiv, quel que soit le profil courant et ce pour l'excution ou l'affichage, mme dans les crans de permissions.

La dfinition d'une `ServiceActivationRule` s'effectue ainsi :

1. Cration d'une rgle sous la forme d'une classe Java implmentant l'interface `ServiceActivationRuleForDataspace`[API] ou `ServiceActivationRuleForDataset`[API], selon le type de service.

2. Assignation de cette rgle aux services concerns au niveau de leur dclaration, selon le type de service via les mthodes `ActivationContextOnDataspace.setActivationRule`[API] ou `ActivationContextWithDatasetSet.setActivationRule`[API].

   La rgle ainsi assigne sera value lors de l'valuation de l'activation du service. Voir Rsolution de permissions pour les services [p 311] pour plus d'informations.

Dfinition des rgles de permission sur service

Les `ServicePermissionRules` sont des rgles avances permettant de dfinir dynamiquement les conditions d'affichage et d'excution d'un service en fonction du contexte (session courante, entit slectionne, etc.). Le service doit au pralable tre activ pour le contexte courant pour que ce type de rgle se dclenche.

La dfinition d'une `ServicePermissionRule` s'effectue de la faon suivante :

1. Cration d'une rgle sous la forme d'une classe Java implmentant l'interface `ServicePermissionRule`[API].
2. Assignation de cette rgle aux services concerns :
   - Soit, pour les nouveaux services, au niveau de leur dclaration via la mthode `ActivationContext.setPermissionRule`[API].

     La rgle ainsi assigne est dite "globale", et est excute ds lors que le service est activ pour le contexte courant. Voir Rsolution de permissions pour les services [p 311] pour plus d'informations.
   - Soit, pour les services existants, dans **l'extension du schma** `SchemaExtensions`[API] via les mthodes `SchemaExtensionsContext.setServicePermissionRuleOnNode`[API] et `SchemaExtensionsContext.setServicePermissionRuleOnNodeAndAllDescendants`[API]. Il est ainsi possible d'assigner une rgle n'importe quel service, y compris les services standards fournis par EBX, sur un ou plusieurs noeuds du modle de donnes: un noeud table, un noeud d'association, etc.

     La rgle ainsi assigne est dite "locale" et n'est excute que dans le contexte du schma tendu et lorsque le noeud concern est celui spcifi. Voir Rsolution de permissions pour les services [p 311] pour plus d'informations.

> **Attention**
> Une seule `ServicePermissionRule` peut tre dfinie pour chaque noeud du modle. Ainsi, la dfinition d'une nouvelle rgle programmatique remplacera l'existante.

49.4 Rsolution des permissions sur donnes

Rsolution des rgles utilisateur

Les droits d'accs dfinis via l'interface utilisateur sont rsolus sur quatre niveaux : espace de donnes, jeu de donnes, enregistrement (le cas chant) et noeud.

Si un profil est associ des droits d'accs restrictifs un niveau donn, le minimum de tous les droits restrictifs dfinis ce niveau est rsolu. Si aucune restriction n'est dfinie pour ce niveau, le maximum de tous les droits d'accs dfinis pour ce niveau est rsolu.

Lorsqu'une permission restrictive est dfinie pour un profil, elle devient prioritaire sur les autres permissions potentiellement accordes par les autres rles de l'utilisateur. D'une manire gnrale, pour toutes les rgles de permission dfinies par l'utilisateur qui s'appliquent la session utilisateur courante :

- Si certaines rgles avec restrictions sont dfinies, les permissions minimales de ces rgles restreintes sont appliques.
- Si aucune rgle avec restrictions n'est dfinie, les permissions maximales de toutes les rgles correspondantes sont appliques.

Exemples :

tant donn deux profils *P1* et *P2* concernant le mme utilisateur, le tableau suivant prsente les possibilits de rsolution de permission de cet utilisateur un service.

Autorisation de P1	Autorisation de P2	Rsolution de permission du service
Activ	Activ	Activ. Les restrictions ne changent rien.
Dsactiv	Dsactiv	Dsactiv. Les restrictions ne changent rien.
Activ	Dsactiv	Activ, moins que l'autorisation de P2 ne soit une restriction.
Dsactiv	Activ	Activ, moins que l'autorisation de P1 ne soit une restriction.

Cette politique de restriction est identique pour les droits d'accs aux donnes.

Dans un autre exemple, un espace de donnes peut tre masqu pour tous les utilisateur en dfinissant une association restrictive entre le profil intgr "Profile.EVERYONE" et le droit d'accs "masqu".

Quel que soit le niveau, les droits d'accs les plus restrictifs entre ceux rsolus ce niveau et ceux rsolus un niveau plus lev sont appliqus. Par exemple, si les permissions d'accs au jeu de donnes d'un utilisateur sont rsolues en accs lecture-criture, mais que l'espace de donnes qui le contient n'autorise que l'accs en lecture, l'utilisateur ne bnficiera que de l'accs en lecture seule pour ce jeu de donnes.

> **Note**
>
> Le mcanisme d'hritage de jeu donnes s'applique aux valeurs et aux droits d'accs. Plus prcisment, les droits d'accs dfinis sur un jeu de donnes seront appliqus sur ses jeux de donnes enfants. Il est possible de surcharger ces droits dans le jeu de donnes enfant.

Exemple de rsolution des droits d'accs

Dans cet exemple, trois utilisateurs appartiennent aux rles dfinis et profils suivants:

Utilisateur	Profil
Utilisateur 1	• utilisateur1 • rle A • rle B
Utilisateur 2	• utilisateur2 • rle A • rle B • rle C
Utilisateur 3	• utilisateur3 • rle A • rle C

Les droits d'accs des profils pour un lment donn sont les suivants :

Profil	Droits d'accs	Politique de restriction
utilisateur1	Masqu	Oui
utilisateur3	Lecture	Non
Rle A	Lecture/criture	Non
Rle B	Lecture	Oui
Rle C	Masqu	Non

Aprs la rsolution base sur les droits d'accs du rle et du profil ci-dessus, les droits appliqus chaque utilisateur sont les suivants:

Utilisateur	Droits d'accs rsolus
Utilisateur 1	Masqu
Utilisateur 2	Lecture
Utilisateur 3	Lecture/criture

Rsolution des droits d'accs aux espaces de donnes et aux images

Au niveau de l'espace de donnes, les droits d'accs sont rsolus de la manire suivante :

- Si un utilisateur a plusieurs droits dfinis via plusieurs profils :
  - Si les droits incluent des restrictions, le minimum des associations restrictives profil-droits est appliqu.
  - Autrement, le maximum des associations profil-droits est appliqu.
- Si l'utilisateur n'a pas de droits dfinis :
  - Si l'utilisateur est un administrateur ou le propritaire de l'espace de donnes, l'accs en lecture-criture est accord pour cet espace de donnes.
  - Autrement, l'espace de donnes sera masqu.

Rsolution des droits d'accs aux jeux de donnes

Au niveau du jeu de donnes, le mme principe qu'au niveau de l'espace de donnes s'applique. Aprs la rsolution des droits d'accs au niveau du jeu de donnes seul, les droits d'accs dfinitifs sont dtermins en prenant les droits les plus bas entre les droits rsolus de l'espace de donnes et les droits rsolus du jeu de donnes. Par exemple, si un espace de donnes est rsolu en lecture seulement pour un utilisateur et qu'un de ses jeux de donnes est rsolu en lecture-criture, l'utilisateur ne bnficiera que de l'accs en lecture seule pour ce jeu de donnes.

Rsolution des droits d'accs aux noeuds

Au niveau du noeud, le mme principe qu'aux niveaux de l'espace de donnes et du jeu de donnes s'applique. Aprs la rsolution des droits d'accs au niveau du noeud seul, les droits d'accs dfinitifs sont dtermins en prenant les droits les plus bas entre les droits rsolus du jeu de donnes et les droits rsolus du noeud.

Des droits d'accs spcifiques peuvent tre dfinis au niveau du noeud. Si aucun droit d'accs spcifique n'est dfini, le droit d'accs par dfaut est utilis pour le processus de rsolution.

> **Note**
>
> La procdure de rsolution diffre lgrement entre les tables et leurs noeuds enfants.

Cas particulier des tables et noeuds de tables

Cette section dcrit le processus de rsolution utilis pour un noeud de table donn ou un enregistrement *N* de la table.

Pour chaque rgle de permission dfinie par l'utilisateur correspondant un des profils de l'utilisateur, les droits d'accs pour *N* seront soit :

1. Les droits d'accs dfinis localement pour *N* ;
2. Hrits des droits d'accs dfinis dans le noeud de table ;
3. Hrits des droits d'accs par dfaut pour les valeurs du jeu de donnes.

Toutes les rgles de permission correspondantes dfinies par l'utilisateur sont utilises pour rsoudre les droits d'accs pour *N*. La rsolution est effectue en fonction de la politique de restriction [p 307].

Les droits d'accs rsolus dfinitifs seront les plus bas entre l'espace de donnes, le jeu de donnes et le droit d'accs rsolu pour *N*.

Rsolution des rgles programmatiques

Il existe trois niveaux de rsolution pour les rgles de droit d'accs programmatiques:le jeu de donnes, l'enregistrement et le noeud. Puisqu'une seule rgle d'accs programmatique peut tre configure pour un niveau donn, la dernire rgle configure sera celle utilise par la procdure de rsolution.

Rsolution de rgle sur jeu de donnes

Pour un jeu de donnes, la dernire rgle est considre comme la rgle rsolue.

Rsolution de rgle sur enregistrement

Pour un enregistrement, la rgle rsolue est la plus basse entre la rgle rsolue configure sur le jeu de donnes et la rgle configure sur cet enregistrement. Voir `SchemaExtensionsContext.setAccessRuleOnOccurrence`[API] pour plus d'informations.

Rsolution de rgle sur noeud

Pour un noeud enfant d'un enregistrement, la rgle rsolue est la plus basse entre la rgle rsolue sur l'enregistrement et la rgle configure pour ce noeud.

Pour un noeud enfant d'un jeu de donnes, la rgle rsolue est la plus basse entre la rgle rsolue configure sur le jeu de donnes et la rgle configure sur ce noeud. Voir `SchemaExtensionsContext.setAccessRuleOnNode`[API] pour plus d'informations.

Politique d'affichage pour les menus droulant des cls trangres

Si un enregistrement est masqu du fait des rgles d'accs, il n'apparaîtra pas dans le menu droulant des cls trangres.

> **Attention**
> Les droits d'accs rsolus sur un jeu de donnes ou un noeud de jeu de donnes sont les droits minimum entre les droits d'accs rsolus dfinis dans l'interface utilisateur et les rgles programmatiques rsolues, le cas chant.

49.5 Rsolution de permissions pour les services

Les services utilisateurs permettent d'excuter des fonctions spcifiques et avances partir de l'interface utilisateur. Selon leur dfinition, ces services peuvent tre appels depuis un menu, comme action dans un workflow, comme item de perspective, ou encore tre excuts directement depuis une URL comme composant web [p 215].

Voir aussi*Overview [p 587]*

Les permissions d'un service sont rsolues ds lors que celui-ci est sollicit dans l'interface utilisateur, c'est dire :

- Lors de son excution, juste avant l'affichage du service.

  Si la permission rsolue dans le contexte pour l'utilisateur n'est pas `enabled`, un message d'interdiction s'affiche la place du service.
- Lors de l'affichage des menus si le service est dfini comme affichable dans les menus.

Si la permission rsolue dans le contexte pour l'utilisateur n'est pas `enabled`, le service n'apparaît pas dans le menu.

Ainsi, chaque sollicitation la rsolution de permissions d'un service se droule de la manire suivante, dans l'ordre et tant que les conditions sont respectes :

1. L'activation du service doit correspondre au contexte courant. Cette activation tient compte :
   - du type d'entit slectionne (jeu de donnes, table, enregistrement, etc.);
   - des rgles statiques d'activation dfinies au sein de la mthode `UserServiceDeclaration.defineActivation`[API];
   - de l'ventuelle rgle dynamique d'activation (ServiceActivationRule [p 306]) galement dfinie au sein de la mthode `UserServiceDeclaration.defineActivation`[API].
2. Lorsque le service est activ pour le contexte courant, les permissions pour la session utilisateur sont alors values :
   - Si des permissions ont t dfinies via l'interface utilisateur pour l'utilisateur courant (ou ses rles), leur rsolution doit retourner `enabled`.

     Pour plus d'informations, se reporter la section Rsolution des rgles utilisateur [p 312] ci-aprs.
   - Si une rgle de permission globale [p 307] est dfinie pour le service, elle doit retourner `enabled` pour le contexte fourni (voir `ServicePermissionRuleContext`[API]).
   - Si une rgle de permission locale [p 307] est dfinie pour le noeud slectionn, elle doit retourner `enabled` pour le contexte fourni (voir `ServicePermissionRuleContext`[API]).

Rsolution des rgles utilisateur

Exemple

Dans cet exemple, nous avons deux utilisateurs appartenant diffrents rles et profils :

Utilisateur	Profils
Utilisateur 1	• utilisateur1 • rle A • rle B
Utilisateur 2	• rle C • rle D

Les permissions associes aux rles et aux profils au niveau du jeu de donnes sont les suivantes :

Profil	**Service prdfini de cration** (`@creation`)	**Service prdfini de duplication** (`@duplicate`)	**Service prdfini de comparaison** (`@compare`)	**Service personnalis 1** (`custom1`)	**Service personnalis 2** (`custom2`)	**Politique de restriction**
utilisateur1	Activ	Dsactiv	Activ	Dsactiv	Activ	Non
Rle A	Activ	Activ	Dsactiv	Activ	Dsactiv	Oui
Rle B	Activ	Dsactiv	Activ	Activ	Dsactiv	Oui
Rle C	Activ	Activ	Dsactiv	Dsactiv	Dsactiv	Non
Rle D	Activ	Dsactiv	Dsactiv	Activ	Dsactiv	Non

Les services disponibles pour chaque utilisateur aprs la rsolution des droits sont les suivantes :

Utilisateurs	**Services disponibles**
Utilisateur 1	Service prdfini de cration (`@creation`)
	Service personnalis 1 (`custom1`)
Utilisateur 2	Service prdfini de cration (`@creation`)
	Service prdfini de duplication (`@duplicate`)
	Service personnalis 1 (`custom1`)

Voir aussi*Rsolution des rgles utilisateur [p 307]*

49.6 Rsolution de permissions pour les actions

Les actions sont les oprations bas-niveau de manipulation d'objets EBX sur lesquels il est possible de spcifier les droits d'excution pour un profil. A la diffrence des permissions sur services utilisateurs, qui ne concernent que l'interface utilisateur, ces droits sont galement applicables lorsqu'une opration est effectue programmatiquement (c'est dire via une `Procedure`[API]) ou indirectement (par exemple lors d'un import de donnes, les actions sur table (cration, surcharge, occultation et suppression) sont values).

Voici la liste des actions sur lesquelles des droits sont dfinissables :

Objet de l'action	Actions disponibles
Espace de donnes	Crer un espace de donnes enfant
	Crer une image
	Initier une fusion
	Exporter une archive
	Importer une archive
	Fermer l'espace de donnes
	Fermer l'image
	Crer un jeu de donnes
Jeu de donnes	Dupliquer le jeu de donnes
	Supprimer le jeu de donnes
	Activer/dsactiver le jeu de donnes
	Crer une vue
Table	Crer un nouvel enregistrement
	Surcharger des enregistrements
	Occulter des enregistrements
	Supprimer des enregistrements

Pour la rsolution des permissions sur actions, seules les permissions dfinies via l'interface utilisateur pour l'utilisateur courant (ou ses rles) sont prises en compte, la politique de restriction s'appliquant comme pour toute permission dfinie via l'interface utilisateur.

Pour plus d'informations, se reporter la section Rsolution des rgles utilisateur [p 315] ci-aprs.

Rsolution des rgles utilisateur

Exemple

Dans cet exemple, nous avons deux utilisateurs appartenant diffrents rles et profils :

Utilisateur	Profils
Utilisateur 1	• utilisateur1 • rle A • rle B
Utilisateur 2	• rle C • rle D

Les droits associs aux rles et aux profils sur les actions d'une table donne sont les suivants :

Profil	Crer un enregistrement	Surcharger un enregistrement	Occulter un enregistrement	Supprimer un enregistrement	Politique de restriction
utilisateur1	Non	Oui	Non	Oui	Non
Rle A	Oui	Non	Oui	Non	Oui
Rle B	Non	Oui	Oui	Non	Oui
Rle C	Oui	Non	Non	Non	Non
Rle D	Non	Non	Oui	Non	Non

Les actions disponibles pour chaque utilisateur aprs la rsolution des droits sont les suivantes :

Utilisateurs	Actions disponibles
Utilisateur 1	Occulter un enregistrement
Utilisateur 2	Crer un enregistrement
	Occulter un enregistrement

Voir aussi*Rsolution des rgles utilisateur* *[p 307]*

CHAPITRE 50

diteur de critres

Ce chapitre contient les sections suivantes :

1. Prsentation
2. Blocs conditionnels
3. Critres atomiques

50.1 Prsentation

L'diteur de critres est prsent dans diffrentes sections de l'interface utilisateur. Il permet de dfinir des filtres pour les tables, ainsi que des rgles de validation et de calcul de donnes. Cet diteur est bas sur la Recommandation W3C XPath 1.0.

Deux types de critres existent:les critres atomiques et les blocs conditionnels.

Voir aussi*Syntaxe XPath supporte* *[p 249]*

50.2 Blocs conditionnels

Les blocs conditionnels sont constitus de critres atomiques et d'autres blocs conditionnels. Ils expriment une condition base sur des critres. Les types de blocs suivants existent :

- **Tous les critres sont faux**: Aucun des critres du bloc n'est vrai.
- **Un critre au moins est faux**: Au moins un critre dans le bloc est faux.
- **Tous les critres sont vrais**: Tous les critres dans le bloc sont vrais.
- **Un critre au moins est vrai**: Un ou davantage de critres sont vrais.

50.3 Critres atomiques

Un prdicat atomique est dfini par un champ, un oprateur et une expression (soit une valeur ou une formule XPath).

Champ	Spcifie le champ de la table sur lequel le critre s'applique.
Oprateur	Spcifie l'oprateur utilis. Les oprateurs disponibles dpendent du type de donnes du champ.
Valeur	Spcifie la valeur ou l'expression. Voir Expression [p 318] ci-dessous.
Code uniquement	Lorsqu'il est coch, il spcifie la recherche des valeurs sous-jacentes pour le champ au lieu des libells, recherchs par dfaut.

Expression

L'expression peut tre une valeur fixe ou une formule. Lors de la cration d'un filtre, seules les valeurs fixes sont autorises. Au cours de la cration d'une rgle de validation ou de calcul, une formule peut tre cre l'aide de l'assistant.

Limitation connue : Le champ de la formule ne valide pas les valeurs en entre, seuls la syntaxe et les chemins sont vrifis.

CHAPITRE **51**

Instructions relatives aux performances

Ce chapitre contient les sections suivantes :

1. Principales vrifications sur les performances
2. Vrifications pour l'utilisation d'un espace de donnes
3. Gestion de la mmoire
4. Validation
5. Mises jour de masse
6. Accs aux tables

51.1 Principales vrifications sur les performances

Bien qu'TIBCO EBX soit conu pour prendre en charge de grands volumes de donnes, plusieurs facteurs courants peuvent engendrer de faibles performances. Les points cl abords dans cette section permettront de rsoudre les ventuels problmes de performance.

Extensions programmatiques coteuses

Pour information, la table ci-dessous dtaille les extensions programmatiques qui peuvent tre implmentes.

Cas d'utilisation	Extensions programmatiques pouvant tre impliques
Validation	• **contraintes programmatiques** `Constraint`[API] • **valeurs calcules** `ValueFunction`[API]
Accs table	• **rgles de permission au niveau de l'enregistrement** `SchemaExtensionsContext.setAccessRuleOnOccurrence`[API] • **filtres programmatiques** `AdaptationFilter`[API]
Affichage du contenu EBX	• **valeurs calcules** `ValueFunction`[API] • **Composants UI** `UIBeanEditor`[API] • **rgles de permission au niveau du noeud** `SchemaExtensionsContext.setAccessRuleOnNode`[API]
Mise jour des donnes	• **triggers** `Package com.orchestranetworks.schema.trigger`[API]

Pour les grands volumes de donnes, les algorithmes lourds ont un impact important sur les performances. Par exemple, la complexit d'un algorithme de contrainte est $O(n^2)$. Si la taille des donnes est 100, le cot rsultant sera proportionnellement de 10 000 (produisant gnralement un rsultat immdiat). Cependant, si la taille des donnes est 10 000, le cot rsultant sera proportionnellement de 10 000 000.

Une autre cause de baisse des performances est l'appel des ressources externes. L'utilisation d'un cache local rsout gnralement ce type de problmes.

Si l'un des cas d'utilisation ci-dessus mne une baisse des performances, il est recommand d'identifier le problme soit via l'analyse du code, soit en utilisant un outil de profiling Java.

Intgration de l'annuaire

L'authentification et la gestion des permissions font appel l'annuaire des rles et utilisateurs [p 423].

Si l'implmentation spcifique d'un annuaire est dploye et accde un annuaire externe, il peut tre utile de mettre en place un cache local. En particulier, une des mthodes les plus frquemment appeles est `Directory.isUserInRole`[API].

Listes agrges

Dans un modle de donnes, lorsque la contrainte de cardinalit d'un lment `maxOccurs` est suprieure 1 et qu'aucune `osd:table` n'est dclare pour cet lment, elle est implmente en tant que `List` Java. Ce type d'lment est appel liste agrge [p 514], par opposition une table.

Il est important de prendre en compte qu'il n'existe aucune optimisation spcifique lors de l'accs aux listes agrges en termes d'itrations, d'affichage utilisateur, etc. En dehors des questions de performance,

les listes agrges sont limites en ce qui concerne les nombreuses fonctionnalits prises en charge par les tables. Voir introduction aux tables [p 517] pour une liste de ces fonctionnalits.

Attention

Pour les raisons indiques ci-dessus, les listes agrges devraient tre utilises uniquement sur de petits volumes de donnes simples (une ou deux douzaines d'enregistrements), sans exigence spcifique concernant leur identification, rsolution, permissions, etc. Pour des volumes de donnes plus importants (ou des fonctionnalits plus avances), il est conseill d'utiliser les dclarations `osd:table`.

51.2 Vrifications pour l'utilisation d'un espace de donnes

Les espaces de donnes [p 98] disponibles en mode smantique sont des outils prcieux pour grer les cycles de vie des donnes complexes. Bien que cette fonctionnalit apporte une grande flexibilit, elle implique aussi un cot supplmentaire, qui doit tre pris en compte pour l'optimisation des processus mtiers.

Cette section rsume les problmes de performance les plus courants qui peuvent apparaître dans le cas d'une utilisation intensive de plusieurs espaces de donnes contenant de grandes tables;et de quelle faon les viter.

Note

Parfois, l'utilisation d'espaces de donnes n'est pas strictement ncessaire. Citons le cas extrme o chaque transaction dclencherait les actions suivantes :

1. Un espace de donnes est cr.
2. La transaction modifie certaines donnes.
3. L'espace de donnes est fusionn, ferm, puis supprim.

Dans ce cas, aucune rfrence l'espace de donnes ne sera ncessaire par la suite;son utilisation pour modifier de donnes isoles n'est donc pas ncessaire. Ainsi, l'utilisation de `Procedure`[API] fournit dj une isolation suffisante pour viter les conflits entre oprations concurrentes. Il serait alors plus efficace de raliser les modifications directement dans l'espace de donnes cible, et de ne pas conserver les tapes de branche et de fusion.

Voici une analogie l'attention des dveloppeurs, portant sur un outil de gestion de code source (CVS, SVN, etc.) : pour raliser une modification simple n'impactant que quelques fichiers, il suffit de le faire directement dans la branche principale. En fait, ce ne serait ni pratique ni durable, en ce qui concerne le tag/la copie des fichiers, si chaque modification de fichier impliquait de brancher l'ensemble du projet, de modifier les fichiers, puis de fusionner la branche ddie.

Mmoire insuffisante

Lorsqu'une table est en mode smantique (par dfaut), le cache de la mmoire Java d'EBX est utilis. Cela garantit un accs aux donnes bien plus efficace lorsque ces donnes sont dj charges dans le cache. Toutefois, s'il n'y a pas assez d'espace pour les donnes courantes, des rechargements intempestifs de donnes depuis la base de donnes sous-jacente vers le cache de la JVM peuvent avoir un lourd impact sur les performances globales.

Cette surcharge d'changes de mmoire (swap) ne peut se produire que pour les tables dans un espace de donnes ayant une stratgie de chargement la demande [p 323].

Ce type de problme peut tre dtect via le fichier log de monitoring [p 323]. Si cela se produit, diffrentes actions peuvent tre envisages :

- rduire le nombre d'espaces de donnes enfants contenant des tables de grande taille ;
- rduire le nombre d'index spcifiquement dfinis pour les tables de grande taille ;
- utiliser le mode relationnel au lieu du mode smantique;
- ou (de manire vidente) allouer davantage de mmoire, ou optimiser la mmoire utilise par les applications pour les objets non-EBX.

Voir aussi

Gestion de la mmoire [p 322]

Mode relationnel [p 267]

Annulation de transaction

En mode smantique, lorsqu'une transaction a ralis des mises jour dans l'espace de donnes courant et qu'elle est interrompue, les index chargs des tables modifies sont rinitialiss. Si les mises jour d'une table de grande taille sont souvent annules et, qu'en mme temps, les accs cette table sont frquents, alors le travail relatif la reconstruction de l'index provoquera un ralentissement de l'accs la table. Par ailleurs, l'allocation de mmoire induite et l'activit de "garbage collector" peuvent mener une rduction globale des performances.

Voir aussi

Garde fonctionnelle et exceptions `TableTrigger.guardAndException`[API]

`Procedure`[API]

Rorganisation des tables de la base de donnes

Comme avec toute base de donnes, l'insertion et la suppression de volumes de donnes importants peuvent mener une fragmentation des donnes, qui peut dtriorer les performances avec le temps. Pour rsoudre ce problme, la rorganisation des tables impactes de la base de donnes est ncessaire. Voir Monitoring and cleanup of the relational database [p 401].

Une spcificit d' EBX est que la cration d'espaces de donnes et d'images ajoute de nouvelles entres aux tables HTA et ATB. Si l'on constate une baisse des performances, il peut donc tre ncessaire de planifier une rorganisation de ces tables, pour les rfrentiels de taille importante dans lesquels de nombreux espaces de donnes sont crs et supprims.

Voir aussi*Monitoring and cleanup of the relational database [p 401]*

51.3 Gestion de la mmoire

Stratgie de chargement

L'administrateur peut prciser la stratgie de chargement de l'espace de donnes ou de l'image dans ses informations. La stratgie par dfaut est de charger et de dcharger les ressources la demande. Pour les ressources frquemment utilises, une stratgie de *chargement forc* est gnralement recommande.

La table suivante dtaille les modes de chargement disponibles en mode smantique. Le serveur d'applications doit tre redmarr afin de prendre en compte tout changement de stratgie de chargement.

Chargement et dchargement la demande	Dans ce mode par dfaut, chaque ressource d'un espace de donnes est uniquement charge ou construite si ncessaire. Les ressources de l'espace de donnes sont rfrences par le logiciel via la classe Java standard `SoftReference`. Par consquent, chaque ressource peut tre dcharge " la discrtion du "garbage collector" en rponse la demande mmoire". L'avantage principal de ce mode est sa capacit librer de la mmoire lorsque cela est ncessaire. En contrepartie, cela implique un cot de chargement/construction lorsque la ressource accde n'a pas encore t charge depuis le dmarrage du serveur, ou si elle a t dcharge depuis.
Chargement forc	Si la stratgie de chargement forc est active pour un espace de donnes ou une image, ses ressources sont charges de manire asynchrone au dmarrage du serveur. Chaque ressource de l'espace de donnes est conserve en mmoire jusqu' l'arrt du serveur ou la fermeture de l'espace de donnes. Ce mode est particulirement recommand pour les espaces de donnes ayant une longue dure de vie et/ou ceux qui sont frquemment utiliss, savoir tout espace de donnes utilis comme rfrence.
Chargement forc et pr-validation	Cette stratgie est similaire la stratgie de chargement forc, la diffrence tant que le contenu de l'espace de donnes charg ou de l'image sera aussi valid au dmarrage du serveur.

Monitoring

Les indications relatives au chargement/dchargement du cache d'EBX sont fournies par le monitoring de la base de donnes sous-jacente, ainsi que par la catgorie de logging du 'monitoring' [p 375].

Si le nombre d'objets *cleared* et *built* demeure lev sur une longue priode, cela signifie qu'EBX effectue de nombreux chargements et dchargements d'objets dans le cache.

Rglage de la mmoire

La mmoire maximum pouvant tre alloue la JVM est habituellement spcifie en utilisant l'option de ligne de commande Java `-Xmx`. Comme c'est le cas pour tout processus intensif, il est important que la taille spcifie par cette option n'excde pas la mmoire physique disponible, ainsi le processus Java n'utilisera pas le disque pour des changes mmoire (swap) au niveau du systme d'exploitation.

Le rglage du "garbage collector" peut aussi amliorer les performances globales. Ce rglage doit tre adapt au cas d'utilisation ainsi qu' l'environnement spcifique Java Runtime utilis.

51.4 **Validation**

Le mcanisme de validation incrmentale permet d'optimiser le travail ncessaire lors des mises jour. Le processus de validation incrmentale a le comportement suivant :

- Le premier appel au rapport de validation d'un jeu de donnes effectue une validation complte du jeu de donnes. La stratgie de chargement [p 322] peut aussi spcifier qu'un espace de donnes doit tre prvalid au dmarrage du serveur.
- Les mises jour de donnes maintiendront le rapport de validation de manire transparente et asynchrone, dans la mesure o les noeuds mis jour dfiniront des dpendances explicites. Par exemple, les facettes standard et statiques, les contraintes de cl trangre, les facettes dynamiques, les noeuds de slection dfinissent des dpendances explicites.
- Si une mise jour de masse est effectue ou s'il y a trop de messages de validation, le processus de validation incrmentale est interrompu. Le prochain appel au rapport de validation dclenchera alors une validation complte.
- Si une transaction est annule, l'tat de validation du jeu de donnes mis jour est rinitialis. Le prochain appel au rapport de validation dclenchera aussi une validation complte.

Certains noeuds sont systmatiquement revalids, et ce, mme si aucune mise jour n'a eu lieu depuis la dernire validation. Ce sont des noeuds *dpendances inconnues*. Un noeud a des dpendances inconnues si:

- Il dfinit une **contrainte programmatique** `Constraint`[API] dans le mode *dpendances inconnues* par dfaut,
- Il dclare une **valeur calcule** `ValueFunction`[API], ou il dclare une facette dynamique qui dpend d'un noeud qui est lui-mme une **valeur calcule** `ValueFunction`[API].
- Il est un Champs hrits [p 296], ou il dclare une facette dynamique qui dpend d'un noeud qui est lui-mme un Champs hrits [p 296].

Par consquent, pour les tables de grande taille (au del de l'ordre de grandeur 10 [5]), il est recommand d'viter les noeuds ayant des dpendances inconnues (ou au moins d'en minimiser le nombre). Pour les contraintes programmatiques, le dveloppeur peut dfinir deux modes alternatifs qui rduiront de manire drastique les cots de validation incrmentale:mode *dpendance locale* et *dpendances explicites*. Pour plus d'informations, voir **Dpendances et validation** `DependenciesDefinitionContext.dependencies`[API].

> **Note**
>
> Il est possible, pour un utilisateur administrateur, de rinitialiser manuellement le rapport de validation d'un jeu de donnes. Cette option est disponible dans la section du rapport de validation dans EBX.

51.5 **Mises jour de masse**

Les mises jour de masse peuvent inclure plusieurs centaines de milliers d'insertions, de modifications et de suppressions. Ces mises jour sont assez rares (habituellement imports de donnes initiales), ou sont ralises de manire non-interactive (traitements par lot de nuit). Ainsi, l'importance des performances pour ces mises jour est moins critique que pour des oprations interactives ou frquentes. Toutefois, tout comme le traitement par lot classique, elles peuvent donner lieu certains problmes spcifiques.

Mode batch

Pour les tables en mode relationnel, l'implmentation des insertions, mises jours et suppressions s'appuie sur la fonction JDBC de traitement par lot (batch). Pour des procdures fort volume, ceci amliore grandement les performances, en limitant le nombre d'allers-retours entre le serveur d'applications et la base de donnes.

Afin d'exploiter totalement cette fonctionnalit, il est possible d'activer le mode batch sur des procdures fort volume. Voir `ProcedureContext.setBatch`[API]. Ceci dsactive le test d'existence avant insertion d'un enregistrement, et rduit ainsi le nombre de requtes mises vers la base de donnes. Le traitement par lot en est rendu encore plus efficace.

Limites de transaction

Il n'est gnralement pas recommand d'utiliser une seule transaction lorsque le nombre de mises jour atomiques dans la transaction est suprieur l'ordre de grandeur 10^4. Les transactions de taille importante ncessitent beaucoup de ressources, particulirement mmoire, d'EBX ainsi que de la base de donnes sous-jacente.

Afin de rduire la taille de la transaction, il est possible de :

- Dfinir la proprit ebx.manager.import.commit.threshold [p 384]. Cependant, cette proprit n'est utilise que pour les imports d'archive interactifs raliss partir de l'interface utilisateur d'EBX.
- Dfinir explicitement un **commit threshold** `ProcedureContext.setCommitThreshold`[API] l'intrieur de la procdure batch.
- Limiter structurellement le primtre de la transaction en implmentant `Procedure`[API] pour une partie de la tche et l'excuter aussi souvent que ncessaire.

D'autre part, dfinir une taille de transaction trs basse peut aussi altrer les performances cause des tches persistantes qui doivent tre excutes sur chaque commit.

> **Note**
>
> Si des commits intermdiaires posent problme du fait que l'atomicit transactionnelle n'est plus garantie, il est recommand d'excuter la mise jour de masse l'intrieur d'un espace de donnes ddi. Cet espace de donnes sera cr juste avant la mise jour de masse. Si la mise jour ne s'achve pas avec succs, l'espace de donnes doit tre ferm, et il faudra rententer la mise jour aprs avoir corrig la cause de l'chec initial. Si la mise jour aboutit avec succs, l'espace de donnes peut alors tre fusionn sans risque avec l'espace de donnes d'origine.

Triggers

Si ncessaire, les triggers peuvent tre dsactivs en utilisant la mthode `ProcedureContext.setTriggerActivation`[API].

51.6 **Accs aux tables**

Fonctionnalits

On accde gnralement aux tables via EBX et aussi via l'API `Request`[API] et les services de donnes. Cet accs implique un ensemble unique de fonctions, incluant un processus de *rsolution dynamique*. Ce processus a le comportement suivant :

- **Hritage**:L'hritage dans l'arborescence du jeu de donnes prend en compte les enregistrements et les valeurs dfinies dans le jeu de donnes parent, en utilisant un processus rcursif. De mme, dans un jeu de donnes racine, un enregistrement peut hriter de certaines de ses valeurs partir des valeurs par dfaut du modle de donnes, dfinies par l'attribut `xs:default`.
- **Calcul de valeur**:Un noeud dclar en tant que `osd:function` est toujours calcul la vole lorsque l'on accde la valeur. Voir `ValueFunction.getValue`[API].
- **Filtrage**:Un prdicat XPath [p 249], un **filtre programmatique** `AdaptationFilter`[API], ou une **rgle de permission** `SchemaExtensionsContext.setAccessRuleOnOccurrence`[API] niveau enregistrement ncessitent une slection d'enregistrements.
- **Tri**:Un tri des enregistrements qui en rsultent peut tre ralis.

Accs aux tables en mode smantique

Architecture et conception

Afin d'amliorer la vitesse des oprations sur les tables, des index sont grs par le moteur d'EBX.

Les fonctions avances d'EBX, telles que le cycle de vie avanc (images et espaces de donnes), l'hritage de jeu de donnes, et la modlisation souple de schma XML, ont conduit une conception spcialise des mcanismes d'indexation. Cette conception peut tre rsume de la faon suivante :

- Les *index* maintiennent une structure de donnes en mmoire, relative une table complte.
- Un index n'est pas sauvegard et sa cration ncessite le chargement de tous les blocs de table de la base de donnes.

Attention

Un accs plus rapide aux tables est assur si les index sont prts et maintenus dans le cache mmoire. Comme mentionn ci-dessus, il est important pour la Machine Virtuelle Java d'avoir un espace allou suffisant pour qu'elle ne libre pas les index trop rapidement.

Facteurs de performance

L'optimiseur de requtes prfre l'utilisation des index lors du calcul du rsultat d'une requte.

Attention

- Seuls les filtres XPath sont pris en compte pour l'optimisation de l'index.
- Les index des cls non-primaires ne sont pas pris en compte pour les jeux de donnes enfants.

En supposant que les index soient dj construits, les impacts sur les performances sont les suivants :

1. Si la requte n'inclut ni filtrage, ni rgles programmatiques, ni tri, alors l'accs ses premires lignes (celles rcupres par une vue pagine) est presque instantan.
2. Si la requte peut tre rsolue sans tape de tri supplmentaire (c'est le cas s'il n'y a pas de critre de tri, ou si ses critres de tri se rapportent ceux de l'index utilis pour le calcul de la requte), l'accs aux premires lignes d'une table devrait tre rapide. Plus prcisment, cela dpend du cot de l'algorithme de filtrage spcifique excut lors de la rcupration d'au moins 2000 enregistrements.

3. Les deux cas ci-dessus garantissent un temps d'accs indpendant de la taille de la table, tout en fournissant une vue trie par l'index utilis. Si un tri supplmentaire est requis, le temps ncessaire au premier accs dpend de la taille de la table, conformment une fonction `Nlog(N)`, o `N` est le nombre d'enregistrements dans la vue rsolue.

> **Note**
>
> Les requtes pagines ajoutent automatiquement la cl primaire la fin du critre spcifi, afin d'assurer un tri cohrent. Ainsi, les champs de cl primaire devraient aussi tre ajouts la fin de tout index destin amliorer les performances des requtes pagines. Celles-ci incluent les vues tabulaires et hirarchiques, ainsi que les menus droulants pour les cls trangres.

Si les index ne sont pas encore construits, ou ont t dchargs, un temps supplmentaire est ncessaire. Le temps de construction est `O(Nlog(N))`.

L'accs aux blocs de donnes de la table est ncessaire lorsque la requte ne peut pas tre calcule sur un index unique (que ce soit pour rsoudre une rgle, filtrer ou trier), ainsi que pour la construction de l'index. Si les blocs de table ne sont pas prsents en mmoire, un temps supplmentaire est ncessaire afin d'aller les chercher dans la base de donnes.

Il est possible d'obtenir des informations via les logs de catgorie "request" et "monitoring" [p 323].

Autres oprations sur les tables

La cration de nouveaux enregistrements ou l'insertion d'enregistrements dpend de l'index de la cl primaire. Ainsi, une cration devient quasi immdiate si cet index est dj charg.

Accs REST aux tables historises

Les informations de fusion dans la table d'historique (le champ `merge_info`) a un cot d'accs potentiellement lev. Pour amliorer les performances et si le code client n'a pas besoin de ce champ, le paramtre `includeMergeInfo` [p 697] doit tre dfini `false`.

Pour plus d'informations, voir Historique [p 273].

Accs aux tables en mode relationnel

Lors du calcul du rsultat d'une requte, le moteur d'EBX dlgue ce qui suit au SGBDR :

- Le traitement de tous les critres de tri des requtes, en les traduisant en clause `ORDER BY`.
- Dans la mesure du possible, le traitement des filtres de requtes, en les traduisant en clause `WHERE`.

> **Attention**
>
> Seuls les filtres XPath sont pris en compte pour l'optimisation de l'index. Si la requte comprend des filtres non optimisables, les lignes de la table seront rcupres dans la base de donnes et filtres dans la mmoire Java par EBX, jusqu' ce que la taille de page demande soit atteinte. Ceci n'est pas aussi efficace que le filtrage ct base de donnes (en particulier en ce qui concerne les E/S).

L'information sur la requte SQL transmise est enregistre dans les logs de catgorie *persistance*. Voir Configuring the EBX logs [p 375].

Indexation

Afin d'amliorer la vitesse des oprations sur les tables, des index peuvent tre dclars sur une table au niveau du modle de donnes. Cela dclenchera la cration d'un index de la table correspondante en base de donnes.

Lors de la conception d'un index visant amliorer les performances d'une requte donne, les mmes rgles que pour la conception d'un index de base de donnes classique s'appliquent.

Dfinir une taille de rcupration (fetch size)

Afin d'amliorer les performances, la taille de rcupration (fetch size) doit tre dfinie en fonction de la taille attendue pour le rsultat de la requte sur une table. Si aucune taille de rcupration n'est dfinie, la valeur par dfaut sera utilise.

- En mode smantique, la valeur par dfaut est 2000.
- En mode mapp, la valeur par dfaut est attribue par le pilote JDBC:10 pour Oracle et 0 pour PostgreSQL.

> **Attention**
>
> Dans PostgreSQL, la valeur par dfaut de 0 charge le pilote JDBC de rcuprer l'ensemble des rsultats en une seule fois, ce qui pourrait causer une `OutOfMemoryError` lors de la rcupration de grandes quantits de donnes. D'autre part, l'utilisation de fetchSize sur PostgreSQL invalidera les curseurs ct serveur en fin de transaction. Si, dans le mme thread, on rcupre d'abord un rsultat dfini avec un fetchSize et que l'on excute ensuite une procdure qui committe la transaction, alors l'accs au rsultat suivant gnrera une erreur.

Voir aussi

`Request.setFetchSize`[API]

`RequestResult`[API]

Guide d'administration (en anglais)

CHAPITRE 52

Administration overview

The Administration section in TIBCO EBX is the main point of entry for all administration tasks. In this overview are listed all the topics that an administrator needs to master. Click on your topic of interest in order to access the corresponding chapter or paragraph in the documentation.

Ce chapitre contient les sections suivantes :

1. Repository management
2. Disk space management
3. Data model
4. Perspectives
5. Administrative delegation

52.1 Repository management

For storage optimization, it is recommended to maintain a repository (persistence RDBMS) to the necessary minimum. To this end, it is recommended to regularly perform a purge of snapshots and obsolete dataspaces and to consider using a backup file system.

See also Cleaning up dataspaces, snapshots, and history [p 401] and Deleting dataspaces, snapshots, and history [p 402].

It is also possible to archive files of the file system type in order to reduce the storage costs, see EBX monitoring [p 400].

Administration tasks can be scheduled by means of the task scheduler, using built-in tasks, see Task scheduler [p 437].

Object cache

EBX maintains an object cache in memory. The object cache size should be managed on a case by case basis according to specific needs and requirements (pre-load option and pre-validate on the reference dataspaces, points of reference, and monitoring), while continuously monitoring the repository health reports (`./ebxLog/monitoring.log`).

See Gestion de la mmoire [p 322].

Obsolete contents

Keeping obsolete contents in the repository can lead to a slow server startup and slow responsiveness of the interface. It is strongly recommended to delete obsolete content.

For example: datasets referring to deleted data models or undeployed add-on modules. See Deploying and registering TIBCO EBX add-ons [p 393].

Workflow

Cleanup

The workflow history and associated execution data have to be cleaned up on a regular basis.

The workflow history stores information on completed workflows, their respective steps and contexts. This leads to an ever-growing database containing obsolete history and can thus lead to poor performance of the database if not purged periodically. See Workflow history [p 436] for more information.

Email configuration

It is required to configure workflow emails beforehand in order to be able to implement workflow email notifications. See Configuration [p 434] for more information.

52.2 Disk space management

Purge of logs

The log file size will vary according to the log level (and to the selected severity level) and disk space needs to be accordingly managed.

An automatic purge is provided with EBX, allowing to define how many days should log files be stored. After the defined period, log files are deleted.

Any customized management of the purge of logs (backup, archiving, etc.) is the user's responsibility.

```
################################################
## Directory of log files 'ebxFile:'
## This property is used by special appender prefixed
## by 'ebxFile:' (see log section below)
################################################
ebx.logs.directory=${ebx.home}/ebxLog

#################################################################
# Daily rollover threshold of log files 'ebxFile:'
# Specifies the maximum number of backup files for daily rollover of 'ebxFile:' appenders.
# When set to a negative value, backup log files are never purged.
# Default value is -1.
#################################################################
ebx.log4j.appender.ebxFile.backup.Threshold=-1
```

Audit trail

EBX is provided with a default audit trail manager. Any customized management (including purge, backups, etc.) is the user's responsibility.

If the audit trail is unwanted, it is possible to fully deactivate it. See Activating the XML audit trail [p 374] and Audit trail [p 443] for more information.

52.3 Data model

Publication management

The management of publications of embedded data models [p 91]. See Data model administration [p 427] for more information on the management of these publications and the administration tasks that can be performed (delete, import and export).

Refresh data models

It is possible to update the data models that are using XML Schema documents not managed by EBX. See Data model refresh tool [p 495] for more information.

52.4 Perspectives

EBX offers extensive UI customization options. Simplified interfaces (Recommended perspectives) [p 419] dedicated to each profile accessing the system can be parameterized by the administrator. According to the profile of the user logging in, the interface will offer more or less options and menus. This allows for a streamlined work environment.

See Advanced perspective [p 408] for more information.

52.5 Administrative delegation

EBX is provided with the built-in administrator profile by default. An administrator can delegate administrative rights to a non-administrator user, either for specific actions or for all activities.

The administrative delegation is defined under 'Administration' in the global permissions [p 407] profile.

Access to the administration section can be granted to specific profiles via the global permissions in order to delegate access rights on corresponding administration datasets.

If all necessary administrative rights have been delegated to non-administrator users, it becomes possible to disable the built-in 'Administrator' role.

Voir aussi*Configuring the user and roles directory [p 373]*

Installation & configuration

CHAPITRE **53**

Supported environments

Ce chapitre contient les sections suivantes :

1. Browsing environment
2. Supported application servers
3. Supported databases

53.1 Browsing environment

Supported web browsers

The TIBCO EBX web interface supports the following browsers:

Microsoft Edge	Minimum supported version is 44 Compatibility mode is not supported.
Microsoft Internet Explorer 10, 11	Compatibility mode is not supported. **Performance limitations:** page loading with IE10 and IE11 is two times slower. This issue is observed when forms have many input components, and particularly many multi-occurrence groups. **Graphical layout:** graphical rendering in IE10 and IE11 can slightly differ from other browsers (for example, the alignment of some labels, icons and other components can be off by a few pixels).
Mozilla Firefox ESR 68 (see details)	As Mozilla Firefox is updated frequently, Cloud Software Group, Inc. only fully supports version ESR 68. See Mozilla Firefox ESR for more details.
Google Chrome	As Google Chrome is updated frequently and it is not possible to deactivate automatic updates, Cloud Software Group, Inc. only tests and makes the best effort to support the latest version available.

Screen resolution

The minimum screen resolution for EBX is 1024x768.

Refreshing pages

Browser page refresh is not supported by EBX. When a page refresh is performed, the last user action is re-executed, and therefore could cause issues. It is thus imperative to use the action buttons and links offered by EBX instead of refreshing the page.

'Previous' and 'Next' buttons

The 'previous' and 'next buttons of the browser are not supported by EBX. When navigating through page history, an obsolete user action is re-executed, and therefore could cause issues. It is thus imperative to use the action buttons and links offered by EBX rather than the browser buttons.

Zoom troubleshooting

Zooming in or out may cause some minor display issues (for example extra scrollbar or misalignment). Those issues can be fixed by refreshing the screen using the provided navigation links.

Browser configuration

The following features must be activated in the browser configuration, for the user interface to work properly:

- JavaScript
- Ajax
- Pop-ups

Attention

Avoid using any browser extensions or plug-ins, as they could interfere with the proper functioning of EBX.

53.2 Supported application servers

EBX supports the following configurations:

- Java Runtime Environment: JRE 8 or 11, which necessarily includes the limitations specified by the Java Virtual Machine implementation vendor. For example, for JRE and JDK 8, Oracle states that they are "not updated with the latest security patches and are not recommended for use in production". See Oracle Java Archive site.
- Any Java application server that complies with Servlet 3.0 (inclusive) up to 5.0 (exclusive), for example Tomcat 7.0 (inclusive) up to 10.0 (exclusive), WebSphere Application Server 8.5.R5 or higher, WebLogic Application Server 12cR2 or higher, JBoss EAP 6.0 or higher. See Java EE deployment overview [p 349].
- The application server must use UTF-8 encoding for HTTP query strings from EBX. This can be set at the application server level.

For example, on Tomcat, you can set the server to always use the UTF-8 encoding, by setting `URIEncoding` to 'UTF-8' on the `<Connector>` in the `server.xml` configuration file. Alternatively, you can instruct the server to use the encoding of the request body by setting the parameter `useBodyEncodingForURI` to 'true' in `server.xml`.

Attention

- Limitations apply regarding clustering and hot deployment/undeployment:

  Clustering: EBX does not include a cache synchronization mechanism, thus it cannot be deployed into a cluster of active instances. See Technical architecture [p 396] for more information.

  Hot deployment/undeployment: EBX does not support hot deployment/undeployment of web applications registered as EBX modules, or of EBX built-in web applications.

53.3 Supported databases

The EBX repository supports the relational database management systems listed below, with the suitable JDBC drivers. It is important to follow the database vendor recommendations and update policies regarding the database itself, as well as the JDBC driver.

Oracle Database 12c or higher (but excluding 18c).	The distinction of `null` values bears certain limitations. On simple `xs:string` elements, Oracle does not support the distinction between empty strings and `null` values. See Empty string management [p 551] for more information. The user with which EBX connects to the database requires the following privileges: • CREATE SESSION, • CREATE TABLE, • ALTER SESSION, • CREATE SEQUENCE, • A non-null quota on its default tablespace.
PostgreSQL 9.6 or higher.	When using PostgreSQL as the underlying database, a request fetch size must be set, otherwise the JDBC driver will fetch the whole result set at once. This could lead to an `OutOfMemoryError` when retrieving large amounts of data. Also, see this limitation [p 290] regarding the evolution of datamodels in mapped modes. See `Request.setFetchSize`[API]. The user with which EBX connects to the database needs the CONNECT privilege on the database hosting the EBX repository. Other than this, the default privileges on the public schema of this database are suitable.
Amazon Aurora PostgreSQL 2.3 (compatible with PostgreSQL 10.7) or higher.	The comments in the above section for PostgreSQL apply.
Google Cloud SQL for PostgreSQL 9.6 (compatible with PostgreSQL 9.6.20) or higher.	The comments in the above section for PostgreSQL apply.
SAP HANA Database 2.0 or Higher.	When using SAP HANA Database as the underlying database, certain schema evolutions are not supported. It is, for example, impossible to reduce the length of a column; this is a limitation of HANA, as mentioned in the SQL

	reference guide: "For row table, only increasing the size of VARCHAR and NVARCHAR type column is allowed." The SAP Hana JDBC driver uses the local timezone of the JVM to handle timestamp SQL columns. Hence, for the specific use cases described in the section Accs SQL aux donnes en mode relationnel [p 271], the JVM powering the Hana JDBC driver - that is the JVM powering EBX - should be started with the property `user.timezone` set to UTC. This configuration is not free of side effects: for example, the timestamps shown in the EBX logs will be in UTC instead of the local timezone.
Microsoft SQL Server 2012 SP4 or higher.	When used with Microsoft SQL Server, EBX uses the default database collation to compare and sort strings stored in the database. This applies to strings used in the data model definition, as well as data stored in relational and history tables. The default database collation can be specified when the database is created. Otherwise, the collation of the database server is used. To avoid naming conflicts or unexpected behaviors, a case- and accent-sensitive collation must be used as the default database collation (the collation name is suffixed by "CS_AS" or the collation is binary). The default setting to enforce transaction isolation on SQL Server follows a pessimistic model. Rows are locked to prevent any read/write concurrent accesses. This may cause liveliness issues for mapped tables (history or relational). To avoid such issues, it is recommended to activate snapshot isolation on your SQL Server database. The user with which EBX connects to the database requires the following privileges: • CONNECT, SELECT and CREATE TABLE on the database hosting the EBX repository, • ALTER, CONTROL, UPDATE, INSERT, DELETE on its default schema.
Microsoft Azure SQL Database	EBX has been qualified on Microsoft Azure SQL Database v12 (12.00.700), and is regularly tested to verify compatibility with the current version of the Azure database service. When used with Microsoft Azure SQL, EBX uses the default database collation to compare and sort strings stored in the database. This applies to strings used in the data model definition, as well as data stored in relational and history tables. The default database collation can be specified when the database is created. Otherwise, the database engine server collation is used. To avoid naming conflicts or unexpected behaviors, a case- and accent-sensitive collation

	must be used as the default database collation (the collation name is suffixed by "CS_AS" or the collation is binary). The user with which EBX connects to the database requires the following privileges: • CONNECT, SELECT and CREATE TABLE on the database hosting the EBX repository, • ALTER, CONTROL, UPDATE, INSERT, DELETE on its default schema.
H2 v1.3.170 or higher.	H2 is not supported for production environments. The default H2 database settings do not allow consistent reads when records are modified. Relational tables are locked following a pessimistic model. To prevent concurrency issues, it is possible to activate the MVCC feature. Note, however, that the H2 documentation states this feature is not yet fully tested.

For other relational databases, please contact the Support team at https://support.tibco.com.

Attention

In order to guarantee the integrity of the EBX repository, it is strictly forbidden to perform direct modifications to the database (for example, using direct SQL writes), except in the specific use cases described in the section Accs SQL aux donnes en mode relationnel [p 271].

Voir aussi

Repository administration [p 396]

Data source of the EBX repository [p 347]

Configuring the EBX repository [p 371]

CHAPITRE **54**

Java EE deployment

Ce chapitre contient les sections suivantes :

1. Introduction
2. Software components
3. Embedded third-party libraries
4. Required third-party libraries
5. Web applications
6. Deployment details
7. Installation notes

54.1 Introduction

This chapter details deployment specifications for TIBCO EBX on a Java application server. For specific information regarding supported application servers and inherent limitations, see Supported environments. [p 334]

54.2 Software components

EBX uses the following components:

- Library `ebx.jar`
- Embedded [p 342] and required [p 342] third-party Java libraries
- EBX built-in web applications [p 345] and optional custom web applications [p 345]
- EBX main configuration file [p 369]
- EBX repository [p 396]
- Default user and roles directory [p 423], integrated within the EBX repository, or a third-party system (LDAP, RDBMS) for the user authentication

Voir aussi*Supported environments [p 334]*

54.3 Embedded third-party libraries

To increase EBX independence and interoperability, it embeds its own third-party libraries. Even if some of them have been modified, preventing conflicts, others must remain unchanged since they are official Java APIs.

The ones that can produce conflicts are:

- Apache Geronimo JSON
- Javax Activation
- Javax Annotations
- Javax JSON Bind
- Javax SAAJ API
- Javax WS RS
- Javax XML Bind

For more information regarding the versions or the details of the Third-Party Library, please refer to the: `TIB_ebx_5.9.22_license.pdf`.

Since those libraries are already integrated, custom web applications should not include them anew, otherwise linkage errors can occur. Furthermore, they should not be deployed aside from the `ebx.jar` library for the same reasons.

54.4 Required third-party libraries

EBX requires several third-party Java libraries. These libraries must be deployed and be accessible from the class-loader of `ebx.jar`. Depending on the application server and the Java runtime environment being used, these libraries may already be present or may need to be added manually.

Database drivers

The EBX repository requires a database. Generally, the required driver is configured along with a data source, if one is used. Depending on the database defined in the main configuration file, one of the

following drivers is required. Keep in mind that, whichever database you use, the version of the JDBC client driver must be equal to or higher than the version of the database server.

H2	Version 2.1.210 validated. Note that H2 is not supported in production environments. https://www.h2database.com/ If the repository has been created using H2 version 1.x, this process must be applied, to migrate to the H2 v2 format.
Oracle JDBC	Oracle database 12cR2 is validated on their latest patch set update. Determine the driver that should be used according to the database server version and the Java runtime environment version. Download the `ojdbc8.jar` certified library with JDK 8. Oracle database JDBC drivers download.
SQL Server JDBC	SQL Server 2012 SP4 and greater, with all corrective and maintenance patches applied, are validated. Remember to use an up-to-date JDBC driver, as some difficulties have been encountered with older versions. Include the `mssql-jdbc-8.4.1.jre8.jar` or `mssql-jdbc-8.4.1.jre11.jar` library, depending on the Java runtime environment version you use. Download Microsoft JDBC Driver 8.4.1 for SQL Server (zip).
PostgreSQL	PostgreSQL 9.6 and above validated Include the latest JDBC driver version 4.2 released for your database server and Java runtime environment. PostgreSQL JDBC drivers download.

Voir aussi

Data source of the EBX repository *[p 347]*

Configuring the EBX repository *[p 371]*

SMTP and emails

The library for JavaMail 1.5.6 email management is required.

The following libraries are used by email features in EBX. See Activating and configuring SMTP and emails [p 378] for details on the configuration.

- `javax.mail.jar`, version 1.5.6, from August 10, 2016
- `smtp.jar`, version 1.5.6, from August 10, 2016
- `pop3.jar`, version 1.5.6, from August 10, 2016

Voir aussi*JavaMail*

Secure Socket Layer (SSL)

These libraries are required if your web applications use SSL features.

- `jsse.jar`: https://www.oracle.com/java/technologies/jsse-v103-for-cdc-v102.html
- `ibmjsse.jar`: https://www.ibm.com/developerworks/java/jdk/security/

Voir aussi*TIBCO EBX main configuration file* *[p 369]*

Java Message Service (JMS)

When using JMS, version 1.1 or higher is required.

Depending on whether a Java EE application server or a Servlet/Java Server Pages (JSP) implementation is being used, the library required is as follows:

- For an application server based on Java EE (Java Platform Enterprise Edition), the required JMS provider library is available by default. See http://www.oracle.com/technetwork/java/javaee/overview for more information.
- For a Servlet/Java Server Pages (JSP) implementation using Java SE (Java Platform Standard Edition), for example Apache Tomcat, a JMS provider library such as Apache ActiveMQ may need to be added. See http://www.oracle.com/technetwork/java/javase/overview for more information.

> **Note**
>
> In EBX, the supported JMS model is exclusively Point-to-Point (PTP). PTP systems allow working with queues of messages.

Voir aussi*TIBCO EBX main configuration file* *[p 369]*

XML Catalog API

A library holding the XML Catalog API, introduces by the JAVA SE 9, is required if your web applications are running over a Java Runtime Environment 8 or below, except when a WebLogic 12c R2 application server is used. To ease the installation steps, the following library has been bundled aside from `ebx.jar`, in the *EBX CD*.

- `xml-apis-1.4.01.jar`, version 1.4.01, from August 20, 2011

See Installation notes [p 349] for more information.

54.5 Web applications

EBX provides pre-packaged EARs that can be deployed directly if your company has no custom EBX module web applications to add. If deploying custom web applications as EBX modules, it is

recommended to rebuild an EAR containing the custom modules packaged at the same level as the built-in web applications.

Attention

Web application deployment on `/` path context is no more supported. The path context must not be empty nor equals to `/`. Moreover, web applications deployment on paths of different depth is deprecated. Every web application path context must be set on the same path depth.

For more information, see the note on repackaging the EBX EAR [p 350] at the end of this chapter.

EBX built-in web applications

EBX includes the following built-in web applications.

Web application name	Description	Required
`ebx`	EBX entry point, which handles the initialization on start up. See Deployment details [p 346] for more information.	Yes
`ebx-root-1.0`	EBX root web application. Any application that uses EBX requires the root web application to be deployed.	Yes
`ebx-ui`	EBX user interface web application.	Yes
`ebx-manager`	EBX user interface web application.	Yes
`ebx-dma`	EBX data model assistant, which helps with the creation of data models through the user interface. **Note:** The data model assistant requires the `ebx-manager` user interface web application to be deployed.	Yes
`ebx-dataservices`	EBX data services web application. Data services allow external interactions with the EBX repository using the SOAP operations [p 639] and Web Services Description Language WSDL generation [p 631] standards or using the Built-in RESTful services [p 681]. **Note:** The EBX web service generator requires the deployment of the `ebx-manager` user interface web application.	Yes

Custom web applications

It is possible to extend and customize the behavior of EBX by deploying custom web applications which conform to the EBX module requirements.

Voir aussi

Packaging TIBCO EBX modules [p 483]

Declaring modules as undeployed [p 385]

54.6 Deployment details

Introduction

This section describes the various options available to deploy the 'ebx' web application. These options are available in its deployment descriptor (`WEB-INF/web.xml`) and are complemented by the properties defined in the main configuration file.

Attention

For JBoss application servers, any unused resources must be removed from the `WEB-INF/web.xml` deployment descriptor.

Voir aussi

TIBCO EBX main configuration file [p 369]

Supported application servers [p 335]

User interface and web access

The web application 'ebx' (packaged as `ebx.war`) contains the servlet `FrontServlet`, which handles the initialization and serves as the sole user interface entry point for the EBX web tools.

Configuring the deployment descriptor for 'FrontServlet'

In the file `WEB-INF/web.xml` of the web application 'ebx', the following elements must be configured for `FrontServlet`:

`/web-app/servlet/load-on-startup`	To ensure that `FrontServlet` initializes upon EBX start up, the `web.xml` deployment descriptor must specify the element `<load-on-startup>1</load-on-startup>`.
`/web-app/servlet-mapping/url-pattern`	`FrontServlet` must be mapped to the path '`/`'.

Configuring the application server for 'FrontServlet'

- `FrontServlet` must be authorized to access other contexts, such as `ServletContext`.

  For example, on Tomcat, this configuration is performed using the attribute `crossContext` in the configuration file `server.xml`, as follows:

  ```
  <Context path="/ebx" docBase="(...)" crossContext="true"/>
  ```

- When several EBX Web Components are to be displayed on the same HTML page, for instance using iFrames, it may be required to disable the management of cookies due to limitations present in some Internet browsers.

  For example, on Tomcat, this configuration is provided by the attribute `cookies` in the configuration file `server.xml`, as follows:

  ```
  <Context path="/ebx" docBase="(...)" cookies="false"/>
  ```

Data source of the EBX repository

> **Note**
>
> If the EBX main configuration specifies the property `ebx.persistence.url`, then the environment entry below will be ignored by EBX runtime. This option is only provided for convenience; it is always recommended to use a fully-configurable datasource. See Configuring the EBX repository [p 371] for more information on this property.

The JDBC datasource for EBX is specified in the deployment descriptor `WEB-INF/web.xml` of the '`ebx`' web application as follows:

Reserved resource name	Default JNDI name	Description
`jdbc/EBX_REPOSITORY`	Weblogic: `EBX_REPOSITORY` JBoss: `java:/EBX_REPOSITORY`	JDBC data source for EBX Repository. Java type: `javax.sql.DataSource`

Voir aussi

Configuring the EBX repository [p 371]

Rules for the database access and user privileges [p 397]

Mail sessions

> **Note**
>
> If the EBX main configuration does not set `ebx.mail.activate` to 'true', or if it specifies the property `ebx.mail.smtp.host`, then the environment entry below will be ignored by EBX runtime. See SMTP [p 378] in the EBX main configuration properties for more information on these properties.

SMTP and email is declared in the deployment descriptor `WEB-INF/web.xml` of the '`ebx`' web application as follows:

Reserved resource name	Default JNDI name	Description
`mail/EBX_MAIL_SESSION`	Weblogic: `EBX_MAIL_SESSION` JBoss: `java:/EBX_MAIL_SESSION`	Java Mail session used to send emails from EBX. Java type: `javax.mail.Session`

JMS connection factory

> **Note**
>
> If the EBX main configuration does not activate JMS through the property `ebx.jms.activate`, the environment entry below will be ignored by the EBX runtime. See JMS [p 379] in the EBX main configuration properties for more information on this property.

The JMS connection factory is declared in the deployment descriptor `WEB-INF/web.xml` of the '`ebx`' web application as follows:

Reserved resource name	Default JNDI name	Description	Required
`jms/EBX_JMSConnectionFactory`	Weblogic: `EBX_JMSConnectionFactory` JBoss: `java:/EBX_JMSConnectionFactory`	JMS connection factory used by EBX to create connections with the JMS provider configured in the operational environment of the application server. Java type: `javax.jms.ConnectionFactory`	Yes

> **Note**
>
> For deployment on WildFly, JBoss and WebLogic application servers with JNDI capabilities, you must update `EBX.ear` or `EBXForWebLogic.ear` for additional mappings of all required resource names to JNDI names.

JMS for data services

To configure data services to use JMS instead of the default HTTP, you must configure the JMS connection factory [p 348] and the following queues, declared in the `WEB-INF/web.xml` deployment descriptor of the '`ebx`' web application. This is the only method for configuring JMS for data services.

When a SOAP request is received, the SOAP response is optionally returned if the header field `JMSReplyTo` is defined. If so, the fields `JMSCorrelationID` and `JMSType` are retained.

See JMS [p 379] for more information on the associated EBX main configuration properties.

> **Note**
>
> If the EBX main configuration does not activate JMS through the property `ebx.jms.activate`, then the environment entries below will be ignored by EBX runtime. See JMS [p 379] in the EBX main configuration properties for more information on this property.

Reserved resource name	Default JNDI name	Description	Required
`jms/EBX_QueueIn`	Weblogic: `EBX_QueueIn` JBoss: `java:/jms/EBX_QueueIn`	JMS queue for incoming SOAP requests sent to EBX by other applications. Java type: `javax.jms.Queue`	No
`jms/EBX_QueueFailure`	Weblogic: `EBX_QueueFailure` JBoss: `java:/jms/EBX_QueueFailure`	JMS queue for failures. It contains incoming SOAP requests for which an error has occurred. This allows replaying these messages if necessary. Java type: `javax.jms.Queue` **Note:** For this property to be read, the main configuration must also activate the queue for failures through the property `ebx.jms.activate.queueFailure`. See JMS [p 379] in the EBX main configuration properties for more information on these properties.	No

JAR files scanner

To speed up the web applications server startup, the JAR files scanner configuration should be modified to exclude, at least, the `ebx.jar` and `ebx-addons.jar` libraries.

For example, on Tomcat, this should be performed in the `tomcat.util.scan.DefaultJarScanner.jarsToSkip` property from the `catalina.properties` file.

54.7 Installation notes

EBX can be deployed on any Java EE application server that supports Servlet 3.0 up to 5.0 except. The following documentation on Java EE deployment and installation notes are available:

- Installation note for JBoss EAP 7.1.x [p 351]
- Installation note for Tomcat 8.x [p 355]
- Installation note for WebSphere AS 9 [p 359]

- Installation note for WebLogic 12c R2 [p 365]

Attention

- The EBX installation notes on Java EE application servers do not replace the native documentation for each application server.
- These are *not* general installation recommendations, as the installation process is determined by architectural decisions, such as the technical environment, application mutualization, delivery process, and organizational decisions.
- In these examples, no additional EBX modules are deployed. To deploy additional modules, the best practice is to rebuild an EAR with the module as a web application at the same level as the other EBX modules. The web application must declare its class path dependency as specified by the Java™ 2 Platform Enterprise Edition Specification, v1.4:

  J2EE.8.2 Optional Package Support

  (...)

  A JAR format file (such as a JAR file, WAR file, or RAR file) can reference a JAR file by naming the referenced JAR file in a Class-Path header in the Manifest file of the referencing JAR file. The referenced JAR file is named using a URL relative to the URL of the referencing JAR file. The Manifest file is named META-INF/MANIFEST.MF in the JAR file. The Class-Path entry in the Manifest file is of the form:

  ```
  Class-Path: list-of-jar-files-separated-by-spaces
  ```

  In an "industrialized" process, it is strongly recommended to develop a script that automatically builds the EAR, with the custom EBX modules, the EBX web applications, as well as all the required shared libraries.
- In order to avoid unpredictable behavior, the guideline to follow is to avoid any duplicates of `ebx.jar` or other libraries in the class-loading system.
- In case of deployment on Oracle WebLogic server, please refer to the Module structure [p 483] section.

CHAPITRE **55**

Installation note for JBoss EAP 7.1.x

Ce chapitre contient les sections suivantes :

1. Overview
2. Requirements
3. Installation
4. Configuration for EBX
5. Updating EBX Enterprise Application aRchive
6. Deploying EBX
7. Start EBX

55.1 Overview

Attention

- This chapter describes a quick installation example of TIBCO EBX on the JBoss Application Server.
- It does not replace the documentation of this application server.
- These are *not* general installation recommendations, as the installation process is determined by architectural decisions such as the technical environment, application mutualization, delivery process, and organizational decisions.
- The complete description of the components required by EBX is given in the following chapter: Java EE deployment [p 341].
- In order to avoid unpredictable behavior, the guideline to follow is to avoid any duplicates of `ebx.jar` or other libraries in the class-loading system.

- JBoss Application Server installation
- `EBX_HOME` directory configuration: copy `ebx.properties`
- Java Virtual Machine properties configuration
- JNDI entries configuration
- Data source and JDBC provider creation

- EBX.ear application update
- EBX.ear application deployment
- EBX application start

55.2 Requirements

- JBoss Application Server EAP 7.1
- Database and JDBC driver
- EBX CD
- No CDI features in EBX's additional modules (since CDI will be automatically disable)

Voir aussi*Supported environments* *[p 334]*

55.3 Installation

This quick installation example is performed for a Linux operating system.

1. To download JBoss EAP 7.1, please first download `Installer` jar version 7.1.0 from: https://developers.redhat.com/products/eap/download/
2. Run the `Installer` using *java -jar* command line.
   For further installation details, please refer to the documentation .
3. Perform a standard installation:
   1. Select the language and click 'OK',
   2. Accept the License and click 'Next',
   3. Choose the installation path and click 'Next',
   4. Keep the 'Component Selection' as it is and click 'Next',
   5. Enter 'Admin username', 'Admin password' and click 'Next',
   6. On 'Installation Overview' click 'Next',
   7. On 'Component Installation' click 'Next',
   8. On 'Configure Runtime Environment' leave selection as it is and click 'Next',
   9. When 'Processing finished' appear, click 'Next',
   10. Uncheck 'Create shortcuts in the start menu' and click 'Next',
   11. Generate 'installation script and properties file' at JBoss EAP 7.1 installation root path,
   12. Click on 'done'.

55.4 Configuration for EBX

EBX home directory creation and configuration

1. Create the *EBX_HOME* directory, for example `/opt/ebx/home`.

2. Copy from the *EBX CD* the `ebx.software/files/ebx.properties` file to *EBX_HOME*. In our example, we will then have the following text file:

   `/opt/ebx/home/ebx.properties`.

3. Edit the `ebx.properties` file to override the default database if needed. By default, the standalone H2 database is defined. The property key `ebx.persistence.factory` must be uncommented for other supported database and it is required to comment the `h2.standalone` one.

Java Virtual Machine properties configuration

1. Open the `standalone.conf` configuration file, placed in *`JBOSS_HOME`*`/bin` (or `jboss-eap.conf` file placed in *`JBOSS_HOME`*`/bin/init.d` for a running server as a service).
2. Add 'ebx.properties' and 'ebx.home' properties to 'JAVA_OPTS' respectively set with `ebx.properties` file's path and *EBX_HOME* directory's path.

JNDI entries configuration

1. Open the `standalone-full.xml` file placed in *`JBOSS_HOME`*`/standalone/configuration`.
2. Add, at least, the following lines to the `server` tag in `messaging-activemq` subsystem:

```
<connection-factory
    name="jms/EBX_JMSConnectionFactory"
    entries="java:/EBX_JMSConnectionFactory"
    connectors="To Be Defined"/>
<jms-queue
    name="jms/EBX_D3ReplyQueue"
    entries="java:/jms/EBX_D3ReplyQueue"
    durable="true"/>
<jms-queue
    name="jms/EBX_QueueIn"
    entries="java:/jms/EBX_QueueIn"
    durable="true"/>
<jms-queue
    name="jms/EBX_QueueFailure"
    entries="java:/jms/EBX_QueueFailure"
    durable="true"/>
<jms-queue
    name="jms/EBX_D3MasterQueue"
    entries="java:/jms/EBX_D3MasterQueue"
    durable="true"/>
<jms-queue
    name="jms/EBX_D3ArchiveQueue"
    entries="java:/jms/EBX_D3ArchiveQueue"
    durable="true"/>
<jms-queue
    name="jms/EBX_D3CommunicationQueue"
    entries="java:/jms/EBX_D3CommunicationQueue"
    durable="true"/>
```

   Warning: the `connectors` attribute value, from the `connection-factory` element, has to be defined. Since the kind of connectors is strongly reliant on the environment infrastructure, a default configuration can not be provided.

   See configuring messaging for more information.

3. Add, at least, the following line to `mail` subsystem:

```
<mail-session name="mail" debug="false" jndi-name="java:/EBX_MAIL_SESSION"/>
```

Data source and JDBC provider creation

1. After the launch of the JBoss Server, run the management CLI without the use of '--connect' or '-c' argument.

2. Use the 'module add' management CLI command to add the new core module. Sample for PostgreSQL configuration:

```
module add \
  --name=org.postgresql \
  --resources=<PATH_TO_JDBC_JAR> \
  --dependencies=javaee.api,sun.jdk,ibm.jdk,javax.api,javax.transaction.api
```

3. Use the 'connect' management CLI command to connect to the running instance.
4. Register the JDBC driver. When running in a managed domain, be sure to precede the command with '/profile=<PROFILE_NAME>'. Sample for PostgreSQL configuration:

```
/subsystem=\
  datasources/jdbc-driver=\
    postgresql:add(\
      driver-name=postgresql,\
      driver-module-name=org.postgresql,\
      driver-xa-datasource-class-name=org.postgresql.xa.PGXADataSource\
    )
```

5. Define the datasource using the 'data-source add' command, specifying the appropriate argument values. Sample for PostgreSQL configuration:

```
data-source add \
  --name=jdbc/EBX_REPOSITORY \
  --jndi-name=java:/EBX_REPOSITORY \
  --driver-name=postgresql \
  --connection-url=jdbc:postgresql://<SERVER_NAME>:<PORT>/<DATABASE_NAME> \
  --user-name=<PERSISTENCE_USER> \
  --password=<PERSISTENCE_PASSWORD>
```

55.5 Updating EBX Enterprise Application aRchive

1. Copy from *EBX CD* the `ebx.software/webapps/ear-packaging/EBX.ear` file to your working directory.
2. Uncompress the ear archive to add the application's specific required third-party libraries.

   Mail: see SMTP and emails [p 343] for more information.

   SSL: see Secure Socket Layer (SSL) [p 344] for more information.

   JMS: see Java Message Service (JMS) [p 344] for more information.

   XML Catalog API: see XML Catalog API [p 344] for more information.
3. Compress anew the ear archive.

55.6 Deploying EBX

1. Copy EBX.ear into *`JBOSS_HOME`*`/standalone/deployments` folder.

55.7 Start EBX

1. After the launch of the JBoss Server, run the EBX web application: http://localhost:8080/ebx/.
2. At first launch, EBX Wizard [p 391] helps you to configure the default properties of your initial repository.

CHAPITRE 56

Installation note for Tomcat 8.x

Ce chapitre contient les sections suivantes :

1. Overview
2. Requirements
3. Installation
4. Configuration for EBX
5. Deploying EBX
6. Start EBX

56.1 Overview

Attention

- This chapter describes a *quick installation example* of TIBCO EBX on Tomcat Application Server.
- It does not replace the documentation of this application server.
- They are *not* general installation recommendations, as the installation process is determined by architectural decisions, such as the technical environment, application mutualization, delivery process, and organizational decisions.
- Tomcat 10.x is not supported.
- The complete description of the components needed by EBX is given in chapter Java EE deployment [p 341].
- In order to avoid unpredictable behavior, the guideline to follow is to avoid any duplicates of `ebx.jar` or other libraries in the class-loading system.
- The description below uses the variable name `$CATALINA_HOME` to refer to the directory into which you have installed Tomcat, and from which most relative paths are resolved. However, if you have configured Tomcat for multiple instances by setting a `$CATALINA_BASE` directory, you should use `$CATALINA_BASE` instead of `$CATALINA_HOME` for each of these references.

- Create `EBX_HOME` directory: copy `ebx.properties`
- Copy EBX and third-party libraries: add libraries (*jar files*) to Tomcat lib directory

- Configure JVM arguments (Java system properties): create the `JAVA_OPTS` environment variable
- Deploy EBX application: copy all war files to Tomcat webapps directory

56.2 Requirements

- Java SE 8 or 11
- Apache Tomcat 8.x
- Database and JDBC driver
- EBX CD

Voir aussi*Supported environments* *[p 334]*

56.3 Installation

1. To download Tomcat 8.x, choose a core binary distributions from https://tomcat.apache.org/download-80.cgi
2. Run the installer or extract the archive and perform standard installation with default options

56.4 Configuration for EBX

1. Create *EBX_HOME* directory, for example `C:\EBX\home`, or `/home/ebx`
2. Copy from *EBX CD* the `ebx.software\files\ebx.properties` file to *EBX_HOME*. In our example, we will then have the following file:

   `C:\EBX\home\ebx.properties`, or `/home/ebx/ebx.properties`, a text file
3. Edit the `ebx.properties` file to override the default database if needed, by default the standalone H2 database is defined. The property key `ebx.persistence.factory` must be uncommented for other supported databases and it is required to comment the `h2.standalone` one.
4. Copy third-party library files to `$CATALINA_HOME\lib\` (or `$CATALINA_BASE\lib\`) directory. In our example, we will have:

   `$CATALINA_HOME\lib\mail.jar`

   `$CATALINA_HOME\lib\h2.jar` (default persistence factory)

   `$CATALINA_HOME\lib\xml-apis-1.4.01.jar` (coming from the *EBX CD* directory `ebx.software\lib\lib-xml-apis\`)

   The exact description of these components is given in chapter Components [p 341]. Obviously, if those components are already deployed on the class-loading system, they do not have to be duplicated
5. Modify the `$CATALINA_HOME\conf\server.xml` (or `$CATALINA_BASE\conf\server.xml`) file. Add the following line to the <Host> element

   ```
   <Context path="/ebx" crossContext="true" docBase="ebx.war"/>
   ```

   After this modification, we will have:

   ```
   <Host name=...>

   ... ...
   ```

```
<Context path="/ebx" crossContext="true" docBase="ebx.war"/>

... ...

</Host>
```

6. Modify the `$CATALINA_HOME\conf\catalina.properties` (or `$CATALINA_BASE\conf\catalina.properties`) file. Add the following lines to the `tomcat.util.scan.DefaultJarScanner.jarsToSkip` property:

   ```
   ebx.jar,\

   ebx-addons.jar,\
   ```

7. Configure the launch properties

   If our Tomcat is launched by a command in Windows' Command Prompt or Unix shell, we can create another launcher file:

   For Windows, edit the launcher file `%CATALINA_HOME%\bin\startup.bat`, and add the following command lines:

   ```
   set EBX_HOME="<path_to_the_directory_ebx_home>"
   set EBX_OPTS="-Debx.home=%EBX_HOME% -Debx.properties=%EBX_HOME%\ebx.properties"
   set JAVA_OPTS="%EBX_OPTS% %JAVA_OPTS%"
   ```

   or for Linux, edit the launcher file `$CATALINA_HOME/bin/startup.sh`, and add the following command lines:

   ```
   EBX_HOME="<path_to_the_directory_ebx_home>"
   EBX_OPTS="-Debx.home=${EBX_HOME} -Debx.properties=${EBX_HOME}/ebx.properties"
   export JAVA_OPTS="${EBX_OPTS} ${JAVA_OPTS}"
   ```

   (!) Accounts used to launch EBX must have create/update/delete rights on *EBX_HOME* directory.

   Windows users that have installed Tomcat as a service may set Java options through the Tomcat service manager GUI (Java tab).

   Be sure to set options on separate lines in the *Java Options* field of the GUI:

   ```
   -Debx.home=<path_to_the_directory_ebx_home>
   -Debx.properties=<path_to_the_directory_ebx_home>\ebx.properties
   ```

   where <path_to_the_directory_ebx_home> is the directory where we copied `ebx.properties`. In our example, it is `C:\EBX\home`, or `/home/ebx`

56.5 **Deploying EBX**

1. Copy from *EBX CD* the `ebx.software\lib\ebx.jar` file to `$CATALINA_HOME\lib\` (or `$CATALINA_BASE\lib\`) directory. In our example, we will have:

   `$CATALINA_HOME\lib\ebx.jar`

2. Copy from *EBX CD* the war files in ebx.software\webapps\wars-packaging to the `$CATALINA_HOME\webapps\` (or `$CATALINA_BASE\webapps\`) directory. In our example, we will have:

   `$CATALINA_HOME\webapps\ebx.war`: Initialization servlet for EBX applications

   `$CATALINA_HOME\webapps\ebx-root-1.0.war`: Provides a common default module for data models

   `$CATALINA_HOME\webapps\ebx-manager.war`: Master Data Management web application

   `$CATALINA_HOME\webapps\ebx-dataservices.war`: Data Services web application

   `$CATALINA_HOME\webapps\ebx-dma.war`: Data Model Assistant web application

`$CATALINA_HOME\webapps\ebx-ui.war`: User Interface web application

56.6 **Start EBX**

1. After Tomcat launch, run EBX web application: http://localhost:8080/ebx/
2. At first launch, EBX Wizard [p 391] helps you to configure the default properties of your initial repository.

CHAPITRE 57

Installation note for WebSphere AS 9

Ce chapitre contient les sections suivantes :

1. Overview
2. Requirements
3. Installation
4. Configuration for EBX
5. Deploying EBX
6. Start EBX

57.1 Overview

Attention

- This chapter describes a quick installation example of TIBCO EBX on the WebSphere Application Server.
- It does not replace the documentation of this application server.
- These are *not* general installation recommendations, as the installation process is determined by architectural decisions such as the technical environment, application mutualization, delivery process, and organizational decisions.
- The complete description of the components required by EBX is given in the following chapter: Java EE deployment [p 341].
- In order to avoid unpredictable behavior, the guideline to follow is to avoid any duplicates of `ebx.jar` or other libraries in the class-loading system.

- Install the WebSphere Application Server
- Create the `EBX_HOME` directory: copy `ebx.properties`
- Create a new profile by using the 'WebSphere Customization Toolbox'
- Create a data source and JDBC provider
- Configure the Java Virtual Machine
- Install the EBX.ear application

- Start the EBX application

57.2 Requirements

- WebSphere Application Server 9
- Database and JDBC driver
- EBX CD
- No CDI features in EBX's additional modules (since CDI will be automatically disabled)

Voir aussi*Supported environments* *[p 334]*

57.3 Installation

This quick installation example is performed for a Linux operating system.

1. To download WebSphere AS 9, please first download the latest 'Installation Manager' from http://www-01.ibm.com/support/docview.wss?uid=swg27025142
2. Run the 'Installation Manager' and add the following repositories:
   - WebSphere Application Server V9.0:

     `http://www.ibm.com/software/repositorymanager/V9WASBase`
   - WebSphere Application Server Network Deployment V9.0:

     `http://www.ibm.com/software/repositorymanager/V9WASND`
3. Install the 'WebSphere Application Server Network Deployment'

   For further installation details, please refer to the documentation.
4. Run the 'WebSphere Customization Toolbox' and perform a standard installation with default options:
   1. Create profile: click 'Create' then select 'Application Server', and click 'Next'
   2. Profile Creation Options: select 'Advanced profile creation' and click 'Next'
   3. Optional Application Deployment: select those options:
      - Deploy the 'Administrative Console'
      - Deploy the 'Installation Verification Tool' application

      Then click 'Next'
   4. Profile Name and Location: enter a profile name (example: 'EbxAppSrvProfile') and directory

      `/opt/IBM/WebSphere/AppServer/profiles/EbxAppSrvProfile`

      further correspond to `PROFILE_HOME` and click 'Next'
   5. Node and Host Names: enter the node name (example: 'Node1'), the server name (example: 'EbxServer'), the host name (example: 'localhost'), and then click 'Next'
   6. Administrative Security: check 'Enable administrative security' option, enter the user name, the password, and click 'Next'
   7. Security Certificate (part 1): select 'Create a new default personnal certificate' and 'Create a new root signing certificate', and click 'Next'

8. Security Certificate (part 2): keep as default and click 'Next'
9. Port Value Assignment: keep as default and click 'Next'
10. Linux Service Definition: check 'Run the application server process as a Linux service' option, enter the user name (example: 'ebx'), and click 'Next'
11. Web Server Definition: keep as default and click 'Next'
12. Profile Creation Summary: keep as default and click 'Create'
13. Profile Creation Complete: uncheck 'Launch the First steps console' option, and click 'Finish'

57.4 Configuration for EBX

1. Create the *EBX_HOME* directory, for example `/opt/ebx/home`
2. Copy from the *EBX CD* the `ebx.software\files\ebx.properties` file to *EBX_HOME*. In our example, we will then have the following text file:

   `/opt/ebx/home/ebx.properties`.
3. Edit the `ebx.properties` file to override the default database if needed. By default, the standalone H2 database is defined. The property key `ebx.persistence.factory` must be uncommented for other supported database and it is required to comment the `h2.standalone` one
4. Create the `EBX_LIB` directory, for example `/opt/ebx/lib`
5. Copy third-party library files from the *EBX CD*, or from other sources, to the `<EBX_LIB>` directory. In our example, for a PostgreSQL database, we will have:

   `postgresql-X.X.X-driver.jar` (coming from another source than the *EBX CD*).

   `xml-apis-1.4.01.jar` (coming from the *EBX CD* directory `ebx.software/lib/lib-xml-apis/`).

   The exact description of these components is given in the chapter Components [p 341]. If those components are already deployed on the class-loading system, they do not have to be duplicated (ex: javax.mail-1.5.6.jar is already present on the WebSphere Application Server, and the database driver is configured at the data source level).
6. Start the server (for example 'EbxServer'),

   sudo <PROFILE_HOME>/bin/startServer.sh <serverName>

   ```
   cd /opt/IBM/WebSphere/AppServer/profiles/EbxAppSrvProfile
   ```

   ```
   sudo bin/startServer.sh EbxServer.
   ```
7. Connect into the 'WebSphere Integrated Solutions Console' using the user name and password typed during the profile creation (Administrative Security step), enter the following url in the browser:

   https://localhost:9043/ibm/console
8. Create a data source and JDBC provider
   1. On the left menu, go to 'Resources > JDBC > Data Sources', to configure your database access. Choose the jdbc 'Scope' (for example use 'Cell'), and click 'New'
   2. Enter basic data source information:
      - Data source name: `EBX_REPOSITORY`
      - JNDI name: `jdbc/EBX_REPOSITORY`

click on 'Next'

3. Select the JDBC provider: select 'Create new JDBC provider', and click 'Next'
4. Create a new JDBC provider: (example with a PostgreSQL database)
   - Database type: `User-defined`
   - Implementation class name: `org.postgresql.ds.PGConnectionPoolDataSource`
   - Name: `PostgreSQL`

   and click 'Next'
5. Enter database class path information: (example with a PostgreSQL database)
   - Class path: `/opt/ebx/lib/postgresql-X.X.X-driver.jar`

   and click 'Next'
6. Enter database specific properties for the data source: keep as default and click 'Next'
7. Setup security aliases: keep as default and click 'Next'
8. Summary: click 'Finish'
9. Save the master configuration

9. Configure the data source: `jdbc/EBX_REPOSITORY`
   1. Click on 'Data Sources > `EBX_REPOSITORY`'
   2. On the right in the 'Configure additional properties' section, click on 'Additional Properties' and define the database account access:
      - Define `user` value to the according user
      - Define `password` value to the according password
   3. Save the master configuration
   4. Test the connection
10. Configure the Java Virtual Machine
    1. Click on 'Application Servers'
    2. Click on the server name (for example: 'EbxServer')
    3. Click on 'Process definition' under 'Server infrastructure > Java Process Management'
    4. Click on 'Java Virtual Machine' under 'Additional Properties'
    5. Enter in 'Generic JVM arguments' the value:

       ```
       -Debx.properties=/opt/ebx/home/ebx.properties -Debx.home=/opt/ebx/home
       ```
    6. Enter in 'Classpath' the paths to the third-party library files placed in the `<EBX_LIB>` directory except for the JDBC driver
    7. click 'Ok'
    8. Save the master configuration

57.5 **Deploying EBX**

1. Copy from the *EBX CD* the `ebx.software/webapps/ear-packaging/EBX.ear` to the *EBX_HOME* directory. In our example, we will have:

`/opt/ebx/ear/EBX.ear`

2. Connect into the 'WebSphere Integrated Solutions Console' using the user name and password typed during the profile creation (Administrative Security step), enter the following url in the browser:

   https://localhost:9043/ibm/console

3. Click on 'WebSphere enterprise applications' under 'Applications > Application Types'
4. Install the EBX application
   1. New Application: On the right panel, click on 'New Enterprise Application'
   2. Preparing for the application installation: Browse your EBX.ear file, located under `/opt/ebx/ear/EBX.ear`, then click 'Next'
   3. How do you want to install the application?: Select 'Fast Path...', then click 'Next'
   4. Select installation options: keep as default, then click 'Next'
   5. Map modules to servers: select all modules, then click 'Next'
   6. Map resource references to resources: copy the 'Resource Reference' value and paste it in the 'Target Resource JNDI Name' field, for all modules, then click 'Next'
   7. Warnings will appear related to `JNDI:mail/EBX_MAIL_SESSION` and `JNDI:jms/EBX_JMSConnectorFactory`. This behavior is normal since we had not configured these resources. Click 'Continue'
   8. Map resource environment references to resources: Copy the 'Resource Reference' value and paste it to the 'Target Resource JNDI Name' value, for all modules, then click 'Next'
   9. Warnings will appear related to non-available resources. This behavior is normal since we had not configured these resources, then click 'Continue'
   10. Map virtual hosts for Web modules: select all modules and click 'Next'
   11. Summary: keep as default, click 'Finish'
   12. If installation succeeds, it logs 'Application EBX installed successfully', then click 'Save'
5. On the left menu, go to 'Applications > Enterprise Applications'
6. Change EBX application's class loader policy
   1. Click on EBX resource’s name
   2. On the 'configuration' pane, under 'Detail Properties', select 'Class loading and update detection'
   3. Under 'General Properties', change 'Class loader order' to 'Classes loaded with local class loader first (parent last)'
   4. Return to 'Applications > Enterprise Applications'
7. Enterprise Applications: select EBX, and then click 'Start'

   The EBX 'Application status' will be changed into a green arrow

57.6 **Start EBX**

1. After the launch of the WebSphere Server, run the EBX web application: http://localhost:9080/ebx/

2. At first launch, EBX Wizard [p 391] helps you to configure the default properties of your initial repository

CHAPITRE **58**

Installation note for WebLogic 12c R2

Ce chapitre contient les sections suivantes :

1. Overview
2. Requirements
3. Installation
4. Configuration for EBX
5. Deploying EBX
6. Start EBX

58.1 Overview

Attention

- This chapter describes a quick installation example of TIBCO EBX on the WebLogic Server.
- It does not replace the documentation of this application server.
- They are *not* general installation recommendations, as the installation process is determined by architectural decisions, such as the technical environment, application mutualization, delivery process, and organizational decisions.
- The complete description of the components needed by EBX is given in the following chapter: Java EE deployment [p 341].
- In order to avoid unpredictable behavior, the guideline to follow is to avoid any duplicates of `ebx.jar` or other libraries in the class-loading system.

- Install the Java Virtual Machine: Create the JAVA_HOME environment variable
- Install the WebLogic Server
- Create the `EBX_HOME` directory: copy `ebx.properties`
- Create a new domain by using the 'Configuration Wizard'
- Configure the domain: Declare EBX `JAVA_OPTIONS` and copy the database JDBC driver
- Install and configure the JDBC driver
- Deploy the EBX application: install dedicated ear file

58.2 Requirements

- Java SE 8 or 11
- WebLogic Server 12c R2
- Database and JDBC driver
- EBX CD

Voir aussi*Supported environments* *[p 334]*

58.3 Installation

1. To download WebLogic 12c R2, please refer to https://edelivery.oracle.com/osdc/faces/Home.jspx
2. Run the 'Fusion Middleware Configuration Wizard' (`<weblogic_home>/oracle_common/common/bin/config.sh`) and perform a standard installation with default options:
   1. Create Domain: choose 'Create a new domain' and specify the domain home folder, then click 'Next'
   2. Templates: keep as default and click 'Next'
   3. Administrator Account: enter a domain administrator username and password and click 'Next'
   4. Domain Mode and JDK: choose the production mode and your jdk installation home and click 'Next'
   5. Advanced configuration: check 'Administration server' and 'Topology'. That way, we create two independent domain nodes: an administration one and an application one, and click 'Next'
   6. Administration Server: enter your administration node name (for example 'AdminServer') and listen port (by default `7001`), then click 'Next'
   7. Managed Servers: add the application node name (for example 'EbxServer') and listen port (for example `7003`), then click 'Next'
   8. Clusters: keep as default and click 'Next'
   9. Machines: keep as default and click 'Next'
   10. Virtual Targets: keep as default and click 'Next'
   11. Partitions: keep as default and click 'Next'
   12. Configuration Summary: click 'Create'
   13. Configuration Process: click 'Next'
   14. End Of Configuration: click 'Finish'

58.4 Configuration for EBX

1. Create the *EBX_HOME* directory, for example `C:\EBX\home`, or `/home/ebx`
2. Copy from the *EBX CD* the `ebx.software\files\ebx.properties` file to *EBX_HOME*. In our example, we will then have the following file:

`C:\EBX\home\ebx.properties`, or `/home/ebx/ebx.properties`, a text file

3. Edit the `ebx.properties` file to override the default database if needed, by default the standalone H2 database is defined. The property key `ebx.persistence.factory` must be uncommented for other supported databases and it is required to comment the `h2.standalone` one.
4. Configure the launch properties for the *Managed Server* (for example 'EbxServer')

   Edit the `<DOMAIN_HOME>/bin/startManagedWebLogic.sh` script file by adding the following lines:

   ```
   EBX_HOME="<path_to_the_directory_ebx_home>"
   EBX_OPTIONS="-Debx.home=${EBX_HOME} -Debx.properties=${EBX_HOME}/ebx.properties"
   export JAVA_OPTIONS="${EBX_OPTIONS} ${JAVA_OPTIONS}"
   ```
5. Copy third-party library files to the `<DOMAIN_HOME>/lib` directory. In our example, for an H2 standalone data base, we will have:

   `h2.jar` (default persistence factory)

   The exact description of these components is given in chapter Components [p 341]. Obviously, if those components are already deployed on the class-loading system, they do not have to be duplicated (ex: mail.jar and xml-apis-1.4.01.jar are already present in the WebLogic Server).
6. Start the 'Administration server' (for example 'AdminServer'), `<DOMAIN_HOME>/bin/startWebLogic.sh`
7. Launch the 'WebLogic Server Administration Console', enter the following url in the browser:

   http://localhost:7001/console.

   Log in with the domain administrator username and password
8. Click on 'Services > Data sources' in the 'Domain Structure' panel, then click on the 'New > Generic Data Source' button
   1. Type Name: `EBX_REPOSITORY`, JNDI Name: `EBX_REPOSITORY` Database Type: *Your database type* and click 'Next'
   2. Choose your database driver type, and click 'Next'
   3. Uncheck 'Supports Global Transactions', and click 'Next'
   4. Setup your database 'Connection Properties' and click 'Next'
   5. Click 'Test Configuration' and then 'Finish'
   6. Switch on the 'Targets' tab and select all Servers, then click 'Save'
   7. Restart the Administration server (for example 'AdminServer')

      `<DOMAIN_HOME>/bin/stopWebLogic.sh`

      `<DOMAIN_HOME>/bin/startWebLogic.sh`

58.5 Deploying EBX

1. Copy from the *EBX CD* the `ebx.software/webapps/ear-packaging/EBXForWebLogic.ear` to the *EBX_HOME* directory. In our example, we will have:

   `C:\EBX\home\EBXForWebLogic.ear`, or `/home/ebx/EBXForWebLogic.ear`
2. Launch the 'WebLogic Server Administration Console', enter the following url in the browser:

   http://localhost:7001/console
3. Click on 'Lock and Edit' in the 'Change Center' panel

4. Click on 'Deployments' in the 'Domain Structure' panel, and click 'Install'
    1. Install Application Assistant: Enter in 'Path' the application full path to EBXForWebLogic.ear file, located on `C:\EBX\home\`, or `/home/ebx/` folder and click 'Next'
    2. Choose the installation type and scope: Click on 'Install this deployment as an application' and 'Global' default scope and click 'Next'
    3. Select the deployment targets: Select a node name (for example 'EbxServer') from the 'Servers' list and click 'Next'
    4. Optional Settings: keep as default and click 'Finish'
5. Click on 'Activate Changes', on the top left corner. The deployment status will change to 'prepared'
6. Switch to 'Control' tab, select the 'EBXForWebLogic' enterprise application, then click on 'Start' > 'Servicing all requests'
7. Start the application node name (for example 'EbxServer'),

    ```
    <DOMAIN_HOME>/bin/startManagedWebLogic.sh EbxServer http://localhost:7001
    ```

58.6 **Start EBX**

1. After WebLogic Server launch, run the EBX web application: http://localhost:7003/ebx/
2. At first launch, EBX Wizard [p 391] helps you to configure the default properties of your initial repository

CHAPITRE 59

TIBCO EBX main configuration file

Ce chapitre contient les sections suivantes :

1. Overview
2. Setting an EBX license key
3. Setting automatic installation on first launch
4. Setting the EBX root directory
5. Configuring the EBX repository
6. Configuring the user and roles directory
7. Configuring EBX localization
8. Setting temporary files directories
9. Activating the XML audit trail
10. Configuring the EBX logs
11. Activating and configuring SMTP and emails
12. Configuring data services
13. Activating and configuring JMS
14. Configuring distributed data delivery (D3)
15. Configuring REST toolkit services
16. Configuring Web access from end-user browsers
17. Configuring failover
18. Tuning the EBX repository
19. Miscellaneous

59.1 Overview

The EBX main configuration file, by default named `ebx.properties`, contains most of the basic parameters for running EBX. It is a Java properties file that uses the standard simple line-oriented format.

The main configuration file complements the Java EE deployment descriptor [p 346]. Administrators can also perform further configuration through the user interface, which is then stored in the EBX repository.

Voir aussi

Deployment details *[p 346]*

UI administration *[p 407]*

Location of the file

The access path to the main configuration file can be specified in several ways. In order of descending priority:

1. By defining the Java system property 'ebx.properties'. For example, this property can be set by adding the option `-Debx.properties=<filePath>` to the `java` command-line command. See Java documentation.
2. By defining the servlet initialization parameter 'ebx.properties'.

   This standard Java EE setting must be specified in the `web.xml` file of the web application '`ebx`'. EBX accesses this parameter by calling the method `ServletConfig.getInitParameter("ebx.properties")` in the servlet `FrontServlet`.

   See getInitParameter in the Oracle `ServletConfig` documentation.
3. By default, if nothing is specified, the main configuration file is located at `WEB-INF/ebx.properties` of the web application '`ebx`'.

> **Note**
>
> In addition to specifying properties in the main configuration file, it is also possible to set the values of properties directly in the system properties. For example, using the `-D` argument of the `java` command-line command.

Custom properties and variable substitution

The value of any property can include one or more variables that use the syntax `${`*`propertyKey`*`}`, where *`propertyKey`* is either a system property, or a property defined in the main configuration file.

For example, the default configuration file provided with EBX uses the custom property `ebx.home` to set a default common directory, which is then included in other properties.

59.2 Setting an EBX license key

Voir aussi*Initialization and first-launch assistant* *[p 391]*

The license key can be retrieved from the TIBCO eDelivery site.

```
################################################
## EBX® License number
## (as specified by your license agreement)
################################################
ebx.license=paste_here_your_license_key
```

59.3 Setting automatic installation on first launch

Repository can be automatically installed on first startup.

```
#################################################################
## Installation on first launch.
## All values are ignored if the repository is  already installed.
#################################################################
```

```
## Enables repository installation on first startup (default is false).
## If true, property ebx.license should also be set to a valid license.
ebx.install.enabled=true

## Following properties configure the repository. Values are optional and defaults are automatically generated.
ebx.install.repository.id=00275930BB88
ebx.install.repository.label=A Test

## Following properties specify the EBX administrator. These are ignored if a custom directory is defined.
ebx.install.admin.login=admin
ebx.install.admin.firstName=admin
ebx.install.admin.lastName=admin
ebx.install.admin.email=adamin@example.com

## Following property specifies the none encrypted password used for the EBX administrator.
## It is ignored if a custom directory is defined. It cannot be set if property
 ebx.install.admin.password.encrypted is set.
#ebx.install.admin.password=admin

## Following property specifies the encrypted password used for the EBX administrator.
## It is ignored if a custom directory is defined. It cannot be set if property ebx.install.admin.password is
 set.
## Password can be encrypted by using command:
##  java -cp ebx.jar com.orchestranetworks.service.directory.EncryptPassword password_to_encrypt
ebx.install.admin.password.encrypted=8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
```

59.4 Setting the EBX root directory

The EBX root directory contains archives, the XML audit trail and, when the repository is persisted on H2 standalone mode, the H2 database files.

```
################################################
## Path for EBX® XML repository
################################################
ebx.repository.directory=${ebx.home}/ebxRepository
```

59.5 Configuring the EBX repository

Before configuring the persistence properties of the EBX repository, carefully read the section Technical architecture [p 396] in the chapter 'Repository administration'.

The required library (driver) for each supported database is described in the chapter Database drivers [p 342].

Voir aussi

Repository administration [p 396]

Rules for the database access and user privileges [p 397]

Supported databases [p 337]

Data source of the EBX repository [p 347]

Database drivers [p 342]

```
################################################################
## The maximum time to set up the database connection,
## in milliseconds.
################################################################
ebx.persistence.timeout=10000
################################################################
## The prefix to add to all table names of persistence system.
## This may be useful for supporting multiple repositories in the relational database.
## Default value is 'EBX_'.
################################################################
ebx.persistence.table.prefix=

################################################################
## Case EBX® persistence system is H2 'standalone'.
################################################################
ebx.persistence.factory=h2.standalone
ebx.persistence.user=sa
```

```
ebx.persistence.password=

######################################################################
## Case EBX® persistence system is H2 'server mode',
######################################################################
#ebx.persistence.factory=h2.server

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.persistence.url=jdbc:h2:tcp://127.0.0.1/ebxdb
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy


######################################################################
## Case EBX® persistence system is Oracle database.
######################################################################
#ebx.persistence.factory=oracle

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.persistence.url=jdbc:oracle:thin:@127.0.0.1:1521:ebxDatabase
#ebx.persistence.driver=oracle.jdbc.OracleDriver
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy


## Activate to use VARCHAR2 instead of NVARCHAR2 on Oracle; never modify on an existing repository.
#ebx.persistence.oracle.useVARCHAR2=false

######################################################################
## Case EBX® persistence system is SAP Hana
######################################################################
#ebx.persistence.factory=hana

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.persistence.url=jdbc:sap://127.0.0.1:39041
#ebx.persistence.driver=com.sap.db.jdbc.Driver
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy


######################################################################
## Case EBX® persistence system is Microsoft SQL Server.
######################################################################
#ebx.persistence.factory=sqlserver

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.persistence.url= \
#jdbc:sqlserver://127.0.0.1:1036;databasename=ebxDatabase
#ebx.persistence.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy


######################################################################
## Case EBX® persistence system is Microsoft Azure SQL database.
######################################################################
#ebx.persistence.factory=azure.sql

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.persistence.url= \
#jdbc:sqlserver://myhost.database.windows.net:1433;database=ebxDatabase;encrypt=true;\
#trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;
#ebx.persistence.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy


######################################################################
## Case EBX® persistence system is PostgreSQL.
######################################################################
#ebx.persistence.factory=postgresql

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
```

```
#ebx.persistence.url=jdbc:postgresql://127.0.0.1:5432/ebxDatabase
#ebx.persistence.driver=org.postgresql.Driver
#ebx.persistence.user=xxxxxxxxx
#ebx.persistence.password=yyyyyyyy
```

59.6 Configuring the user and roles directory

This parameter specifies the Java directory factory class name. It must only be defined if not using the default EBX directory.

Voir aussi

Users and roles directory *[p 423]*

`DirectoryFactory`[API]

```
################################################
## Specifies the Java directory factory class name.
## Value must be the fully qualified name of the Java class.
## The class must extend com.orchestranetworks.service.directory.DirectoryFactory.
################################################
#ebx.directory.factory=xxx.yyy.DirectoryFactoryImpl
```

It is also possible to disable the built-in role "ADMINISTRATOR".

```
################################################
## Specifies whether the built-in role ADMINISTRATOR is disabled.
## Default value is false.
################################################
#ebx.directory.disableBuiltInAdministrator=true
```

59.7 Configuring EBX localization

This parameter is used to configure the locales used at runtime. This list must contain all the locales that are exposed to the end-user. EBX will not be able to display labels and messages in a language that is not declared in this list.

The default locale must be the first one in the list.

```
#####################################################################
## Available locales, separated by a comma.
## The first element in the list is considered as the default locale.
## If not set, available locales are 'en-US, fr-FR'.
##
#####################################################################
#ebx.locales.available=en-US, fr-FR
```

Voir aussi*Extension de l'internationalisation de TIBCO EBX* *[p 261]*

59.8 Setting temporary files directories

Temporary files are stored as follows:

```
################################################
## Directories for temporary resources.
################################################
# The property ebx.temp.directory allows to specify a directory for temporary files.
# Default value is java.io.tmpdir
#
ebx.temp.directory = \\${java.io.tmpdir}
#ebx.temp.directory = /tmp/java

# The property ebx.temp.cache.directory allows to specify the directory containing temporary files for cache.
# Default value is ${ebx.temp.directory}/ebx.platform.
#ebx.temp.cache.directory = ${ebx.temp.directory}/ebx.platform

# The property ebx.temp.import.directory allows to specify the directory containing temporary files for import.
# Default value is ${ebx.temp.directory}/ebx.platform.
```

```
#ebx.temp.import.directory = ${ebx.temp.directory}/ebx.platform
```

59.9 Activating the XML audit trail

By default, the XML audit trail is activated. It can be deactivated using the following variable:

```
######################################################################
# The XML history has been replaced by an SQL history.
# This old XML history can be deactivated using the following variable.
# Default is true.
######################################################################
ebx.history.xmlaudittrail.activated = true
```

Voir aussi*Audit trail* *[p 443]*

59.10 Configuring the EBX logs

The most important logging categories are:

ebx.log4j.category.log.kernel	Logs for EBX main features, processes, exceptions and compilation results of modules and data models.
ebx.log4j.category.log.workflow	Logs for main features, warnings and exceptions about workflow.
ebx.log4j.category.log.persistence	Logs related to communication with the underlying database.
ebx.log4j.category.log.setup	Logs for the compilation results of all EBX objects, except for modules and data models.
ebx.log4j.category.log.validation	Logs for datasets validation results.
ebx.log4j.category.log.mail	Logs for the activity related to the emails sent by the server (see Activating and configuring SMTP and emails [p 378]). **Note:** This category must not use the Custom SMTP appender [p 376] in order to prevent infinite loops.
ebx.log4j.category.log.d3	Logs for D3 events on EBX.
ebx.log4j.category.log.dataservices	Logs for data service events in EBX.
ebx.log4j.category.log.monitoring	Raw logs for monitoring [p 323].
ebx.log4j.category.log.request	The optimization strategy for every `Request`[API] issued on a semantic table in the EBX repository.
ebx.log4j.category.log.restServices	Logs for REST services events in EBX, including those from the REST Toolkit [p 745].

Some of these categories can also be written to through custom code using the `LoggingCategory`[API] interface.

```
 ################################################
## Log4J properties:
##
## We have some specific syntax extensions:
## - Appender ebxFile:<aFileName>
## Defines a file appender with default settings (threshold=DEBUG)
##
```

```
## - property log.defaultConversionPattern is set by Java
################################################
#ebx.log4j.debug=true
#ebx.log4j.disable=
ebx.log4j.rootCategory= INFO
ebx.log4j.category.log.kernel= INFO, Console, ebxFile:kernel, kernelMail
ebx.log4j.category.log.workflow= INFO, ebxFile:workflow
ebx.log4j.category.log.persistence= INFO, ebxFile:persistence
ebx.log4j.category.log.setup= INFO, Console, ebxFile:kernel
ebx.log4j.category.log.mail= INFO, Console, ebxFile:mail
ebx.log4j.category.log.frontEnd= INFO, Console, ebxFile:kernel
ebx.log4j.category.log.frontEnd.incomingRequest= INFO
ebx.log4j.category.log.frontEnd.requestHistory= INFO
ebx.log4j.category.log.frontEnd.UIComponentInput= INFO
ebx.log4j.category.log.fsm= INFO, Console, ebxFile:fsm
ebx.log4j.category.log.fsm.dispatch= INFO
ebx.log4j.category.log.fsm.pageHistory= INFO
ebx.log4j.category.log.wbp= FATAL, Console
#-----------------------------------------------
ebx.log4j.appender.Console.Threshold = INFO
ebx.log4j.appender.Console=com.onwbp.org.apache.log4j.ConsoleAppender
ebx.log4j.appender.Console.layout=com.onwbp.org.apache.log4j.PatternLayout
ebx.log4j.appender.Console.layout.ConversionPattern=${log.defaultConversionPattern}
#-----------------------------------------------
ebx.log4j.appender.kernelMail.Threshold = ERROR
ebx.log4j.appender.kernelMail = com.onwbp.org.apache.log4j.net.SMTPAppender
ebx.log4j.appender.kernelMail.To = admin@domain.com
ebx.log4j.appender.kernelMail.From = admin${ebx.site.name}
ebx.log4j.appender.kernelMail.Subject = EBX® Error on Site ${ebx.site.name} (VM ${ebx.vm.id})
ebx.log4j.appender.kernelMail.layout.ConversionPattern=**Site ${ebx.site.name} (VM${ebx.vm.id})**%n
${log.defaultConversionPattern}
ebx.log4j.appender.kernelMail.layout = com.onwbp.org.apache.log4j.PatternLayout

#-----------------------------------------------
ebx.log4j.category.log.monitoring= INFO, ebxFile:monitoring
ebx.log4j.category.log.dataServices= INFO, ebxFile:dataServices
ebx.log4j.category.log.d3= INFO, ebxFile:d3
ebx.log4j.category.log.request= INFO, ebxFile:request
ebx.log4j.category.log.restServices= INFO, ebxFile:dataServices
```

Custom 'ebxFile' appender

The token `ebxFile:` can be used as a shortcut to define a daily rolling file appender with default settings. It must be followed by a file name. It then activates an appender that writes to a file located in the directory `ebx.logs.directory`.

The property `ebx.log4j.appender.ebxFile.backup.Threshold` allows defining the maximum number of backup files for daily rollover.

```
################################################
## Directory of log files 'ebxFile:'
## This property is used by special appender prefixed
## by 'ebxFile:' (see log section below)
################################################
ebx.logs.directory=${ebx.home}/ebxLog

#################################################################
# Daily rollover threshold of log files 'ebxFile:'
# Specifies the maximum number of backup files for daily rollover of 'ebxFile:' appenders.
# When set to a negative value, backup log files are never purged.
# Default value is -1.
#################################################################
ebx.log4j.appender.ebxFile.backup.Threshold=-1
```

Custom SMTP appender

The appender `com.onwbp.org.apache.log4j.net.SMTPAppender` provides an asynchronous email sender.

Voir aussi*Activating and configuring SMTP and emails* *[p 378]*

Module's appenders configuration

The configurations of module logging categories are declared according to a hierarchical pattern. Furthermore, every enabled logging request for a given category will be forwarded to all the appenders defined for that category, as well as to the appenders higher in the hierarchy. This fact means that appenders are inherited additively.

The root module logging category can be customized by setting the property `ebx.log4j.category.log.wbp`. For a given module, if the root module logging category configuration is the only one applicable, then a `DailyRollingFileAppender` is automatically added to the module's appenders list. The log file name will be derived from the module's name and from an optional specific sub-category.

```
########################################################################
## Root module logging category configuration
########################################################################
ebx.log4j.category.log.wbp = FATAL, Console
```

Every module logging category can be customized by setting the property `ebx.log4j.category.log.module.xxxxxx`, where `xxxxxx` corresponds to the module's name. The explicit configuration of the module logging category will disable the automatic addition of the `DailyRollingFileAppender` previously described. However, any other defined appenders will be inherited additively.

Since the inheritance mechanism may not be suitable for every case, the additivity can be broken by setting to `false` the property `ebx.log4j.additivity.log.wbp.xxxxxx`, where `xxxxxx` corresponds to the module's name.

```
########################################################################################
## Module logging category configuration
##
## The module's log messages will only be written to mycompany-module log file
## since additivity has been broken
########################################################################################
ebx.log4j.category.log.module.mycompany-module = INFO, ebxFile:mycompany-module

ebx.log4j.additivity.log.wbp.mycompany-module = false
```

Module's log threshold

The inheritance mechanism described in Module's appenders configuration [p 377] is applied to the module log threshold as well. Actually, the inherited level for a given logger, is equal to the first non-null level in the hierarchy, starting from this logger and proceeding up to the root module logging category.

Custom module log threshold

There is an exception to the inheritance, for custom module, since the log level threshold of their logging category, by default, is set to `INFO`. This threshold can be customized by setting the property `ebx.log4j.category.log.module.xxxxxx`, where `xxxxxx` corresponds to the custom module's name.

Example: `ebx.log4j.category.log.module.mycompany-module=DEBUG`.

Voir aussi *`ModuleContextOnRepositoryStartup.getLoggingCategory`*[API]

Add-on module log threshold

Like custom module, by default, the log level threshold of any add-on module is set to `INFO`.

The log level threshold can be customized by setting the property `ebx.log4j.category.log.addon.xxxxxx` where `xxxxxx` corresponds to the add-on module's name.

Example: `ebx.log4j.category.log.addon.daqa=DEBUG`

Modules log threshold overrides summary

Thus and considering EBX logging features, the log threshold defined at the root level in the logger hierarchy may be overridden by (by order from least to most weighted):

- the module logging category explicit configuration,
- the module's `log.threshold.xxxx` or `log.threshold.category.xxxx` log configuration properties, where `xxxx` corresponds to the sub category,
- the following log configuration properties: `ebx.log4j.category.log.module.xxxxxx`, where `xxxxxx` corresponds to the custom module's name, or `ebx.log4j.category.log.addon.xxxxxx`, where `xxxxxx` corresponds to the add-on module's name.

59.11 Activating and configuring SMTP and emails

The internal mail manager sends emails asynchronously. It is used by the workflow engine and the custom SMTP appender `com.onwbp.org.apache.log4j.net.SMTPAppender`.

Voir aussi*Mail sessions* *[p 347]*

```
################################################
## SMTP and emails
################################################

## Activate emails (true or false, default is false).
## If activated, the deployer must ensure that the entry 'mail/EBX_MAIL_SESSION' is bound
## in the operational environment of the application server (except if a specific email
## configuration is used by setting the property ebx.mail.smtp.host below).
#ebx.mail.activate=false

## Polling interval is in seconds (default is 10).
#ebx.mail.polling.interval=10

## Specific properties to be set only only if you want to ignore the standard
## deployment process of 'ebx' web application in the target operational environment
## (see the deployment descriptor 'web.xml' of 'ebx' web application).
#ebx.mail.smtp.host = smtp.domain.com
## SMTP port default is 25.
#ebx.mail.smtp.port= 25
#ebx.mail.smtp.login=
#ebx.mail.smtp.password=
## Activate SSL (true or false, default is false).
## If SSL is activated, a SSL factory and a SSL provider are required.
#ebx.mail.smtp.ssl.activate=true
#ebx.mail.smtp.ssl.provider=com.sun.net.ssl.internal.ssl.Provider
#ebx.mail.smtp.ssl.factory=javax.net.ssl.SSLSocketFactory
```

59.12 Configuring data services

```
#####################################################################
## Data services
#####################################################################

# Specifies the default value of the data services parameter
# 'disableRedirectionToLastBroadcast'.
# Default is false.
#ebx.dataservices.disableRedirectionToLastBroadcast.default=false

# Specifies the default value for deletion at the end of close and
# merge operations.
# If the parameter is set in the request operation, it overrides
# this default setting.
```

```
# If unspecified, default is false.
#ebx.dataservices.dataDeletionOnCloseOrMerge.default=false
#ebx.dataservices.historyDeletionOnCloseOrMerge.default=false

# Upon WSDL generation, specifies if the target namespace value
# corresponds to the content before 5.5.0 'ebx-services'
# or 'urn:ebx:ebx-services' in conformity with the URI syntax.
# If the parameter is set to true, there is no check of the target
# namespace as URI at the WSDL generation.
# If unspecified, default is false.
#ebx.dataservices.wsdlTargetNamespace.disabledCheck=false

######################################################################
## REST configuration
######################################################################

# If activated, the HTTP request header 'Accept' is used to specify
# the accepted content type. If none is supported, an error is
# returned to the client with the HTTP code 406 'Not acceptable'.
# If deactivated, the header is ignored therefore the best content
# type is used.
# Default is false.
#ebx.dataservices.rest.request.checkAccept=false

# If activated, it tries authentication 'Basic Authentication Scheme'
# method and set 'Basic' value in 'WWW-Authenticate' header of HTTP
# response.
# Default is false.
#ebx.dataservices.rest.auth.tryBasicAuthentication=false

# Authorization token timeout is seconds.
# Default value is 1800 seconds (30 minutes)
# This value is ignored if 'Token Authentication Scheme' is not activated.
#ebx.dataservices.rest.auth.token.timeout=1800
```

59.13 Activating and configuring JMS

Voir aussi*JMS for data services* *[p 348]*

```
######################################################################
## JMS configuration for Data Services
######################################################################

## Activates JMS (true or false, default is false).
## If activated, the deployer must ensure that the entry 'jms/EBX_JMSConnectionFactory'
## are bound in the operational environment of the application server.
## The entry 'jms/EBX_QueueIn' should also be bound to enable handling Data Services
## request using JMS.
#ebx.jms.activate=false

## Activates JMS queue for failures (true or false, default is false).
## If activated, the deployer must ensure that the entry 'jms/EBX_QueueFailure' is bound
## in the operational environment of the application server.
#ebx.jms.activate.queueFailure=false

## Number of concurrent listener(s)
## Default is 3.
## Property is used if ebx.jms.activate is set to true.
#ebx.jms.listeners.count=3
```

59.14 Configuring distributed data delivery (D3)

See Configuring D3 nodes [p 467] for the main configuration file properties pertaining to D3.

Voir aussi

JMS for distributed data delivery (D3) *[p 457]*

Introduction to D3 *[p 448]*

59.15 Configuring REST toolkit services

```
######################################################################
## REST configuration
######################################################################

# Defines the maximum number of bytes that will be extracted
# from the REST request body to build some DEBUG log messages.
# Default value is 8192 bytes.
# This value is ignored if DEBUG level is not activated on the restServices logger.
#ebx.restservices.log.body.content.extract.size=8192
```

59.16 Configuring Web access from end-user browsers

HTTP Authorization header policy

EBX natively offers three policies to send and receive credentials using HTTP headers:

`standard`	It corresponds to the authentication scheme, using the HTTP Authorization header, described in the RFC 2617.
`ebx`	To prevent HTTP Authorization header override issues, this policy acts the same as the `standard` but the credentials are stored in an EBX specific HTTP header.
`both`	It is the combination of the two previously described policies.

```
##################################################
## EBX® authorization header policy for HTTP requests
##
## Possible values are: standard, ebx, both.
##  standard:
##    the standard HTTP Authorization header holds the credentials
##  ebx:
##    an EBX® specific HTTP header holds the credentials
##  both:
##    both (standard and specific) HTTP headers hold the credentials
##
## Default value is: both.
##################################################
#ebx.http.authorization.header.policy=both
```

URLs computing

By default, EBX runs in "standalone" mode, where external resources (images, JavaScript, etc.) are provided by the application server.

Also by default, URL-related parameters in the main configuration file do not have to be set.

In this case, the server name and the port are obtained from the initial request sent to EBX.

Voir aussi*URL policy (deprecated)* *[p 410]*

```
######################################################################
## EBX® FrontServlet: default properties for computing servlet address
##
## {useLocalUrl}:
## If set to true, servlet address is a "local absolute" URL.
## (that is, a relative URL consisting of an absolute path: "/path")
```

```
## See RFC 2396, http://www.ietf.org/rfc/rfc2396.txt).
## This property is defined once for HTTP and HTTPS.
## Default value is false.
##
## {host}:
## If neither defined nor adapted, retrieves initial request host
## {port}:
## If neither defined nor adapted, retrieves initial request host
## {path}:
## Mandatory, may be empty
## {ui.path}:
## If not defined, defaults to ebx-ui/
## {http.useHttpsSettings}:
## If true, force the use of SSL security even if the incoming
## requests do not. Default value is false.
##
## Resulting address will be:
## EBX®: protocol://{host}:{port}/{path}
## UI: protocol://{host}:{port}/{ui.path}
##
## Each property for HTTP (except {port}) may be inherited from HTTPS property,
## and reciprocally.
#####################################################################

ebx.servlet.useLocalUrl=true

#ebx.servlet.http.host=
#ebx.servlet.http.port=
ebx.servlet.http.path=ebx/
#ebx.servlet.http.ui.path=ebx-ui/
#ebx.servlet.http.useHttpsSettings=false

#ebx.servlet.https.host=
#ebx.servlet.https.port=
ebx.servlet.https.path=ebx/
#ebx.servlet.https.ui.path=ebx-ui/

#####################################################################
## External resources: default properties for computing external resources address
##
## The same rules apply as EBX® FrontServlet properties (see comments).
##
## Each property may be inherited from EBX® FrontServlet.
#####################################################################

ebx.externalResources.useLocalUrl=true

#ebx.externalResources.http.host=
#ebx.externalResources.http.port=
#ebx.externalResources.http.path=
#ebx.externalResources.http.useHttpsSettings=false

#ebx.externalResources.https.host=
#ebx.externalResources.https.port=
#ebx.externalResources.https.path=
```

Proxy mode

Proxy mode allows using a front-end HTTP server to provide static resources (images, CSS, JavaScript, etc.). This architecture reduces the load on the application server for static HTTP requests. This configuration also allows using SSL security on the front-end server.

The web server sends requests to the application server according to a path in the URL. The `servletAlias` and `uiServletAlias` paths are specified in the main configuration file.

The web server provides all external resources. These resources are stored in a dedicated directory, accessible using the `resourcesAlias` path.

EBX must also be able to access external resources from the file system. To do so, the property `ebx.webapps.directory.externalResources` must be specified.

To force the use of SSL security even if the incoming requests do not, `ebx.servlet.http.useHttpsSettings` and / or `ebx.externalResources.http.useHttpsSettings` properties must be set to `true`. Their default values are `false`.

The main configuration file may be configured as follows:

```
################################################
## Path for external resources if they are not
## delivered within web applications
## This field is mandatory if in proxy mode.
################################################
ebx.webapps.directory.externalResources=D:/http/resourcesFolder

################################################

ebx.servlet.useLocalUrl=true

#ebx.servlet.http.host=
#ebx.servlet.http.port=
ebx.servlet.http.path=servletAlias
ebx.servlet.http.ui.path=uiServletAlias
#ebx.servlet.http.useHttpsSettings=false

#ebx.servlet.https.host=
#ebx.servlet.https.port=
ebx.servlet.https.path=servletAlias
ebx.servlet.https.ui.path=uiServletAlias

################################################

ebx.externalResources.useLocalUrl=true

#ebx.externalResources.http.host=
#ebx.externalResources.http.port=
ebx.externalResources.http.path=resourcesAlias
#ebx.externalResources.http.useHttpsSettings=false

#ebx.externalResources.https.host=
#ebx.externalResources.https.port=
ebx.externalResources.https.path=resourcesAlias
```

Attention

When proxy mode is used, the URL to the `ebx-dataservices` module must be configured through the lineage administration panel. Note that the provided URL must end its path with `/ebx-dataservices`.

Voir aussi*Path recommendations*

Reverse-proxy mode

If URLs generated by EBX, for requests and external resources, must contain a different protocol than the one from the incoming request, a specific server name, a specific port number or a specific path prefix, properties may be configured as follows:

```
################################################
#ebx.servlet.useLocalUrl=false

ebx.servlet.http.host=reverseDomain
#ebx.servlet.http.port=
ebx.servlet.http.path=ebx/
#ebx.servlet.http.ui.path=ebx-ui/
#ebx.servlet.http.useHttpsSettings=false

ebx.servlet.https.host=reverseDomain
#ebx.servlet.https.port=
ebx.servlet.https.path=ebx/
#ebx.servlet.https.ui.path=ebx-ui/

################################################
## Web parameters (for external resources)
## if nothing is set, values are taken from servlet.
################################################
#ebx.externalResources.useLocalUrl=false

#ebx.externalResources.http.host=
#ebx.externalResources.http.port=
#ebx.externalResources.http.path=
ebx.externalResources.http.useHttpsSettings=true
```

```
ebx.externalResources.https.host=reverseDomain
#ebx.externalResources.https.port=
ebx.externalResources.https.path=
```

Attention

When reverse-proxy mode is used, the URL to the `ebx-dataservices` module must be configured through the lineage administration panel. Note that the provided URL must end its path with `/ebx-dataservices`.

Voir aussi*Path recommendations*

Path recommendations

It is recommended to deploy webapps of EBX modules on the same path level. For example:

- `/module1`
- `/module2`

or:

- `/path/module1`
- `/path/module2`

It avoids difficulties when configuring the advanced features of Proxy mode and Reverse-proxy mode.

Consequently, it is not recommended to use the root path `"/"` as the servlet's HTTP or HTTPS path in the following properties:

- `ebx.servlet.http.path`
- `ebx.servlet.http.ui.path`
- `ebx.servlet.https.path`
- `ebx.servlet.https.ui.path`

59.17 Configuring failover

These parameters are used to configure the failover mode and activation key, as well as heartbeat logging in `DEBUG` mode.

Voir aussi*Failover with hot-standby* *[p 397]*

```
################################################
## Mode used to qualify the way in which a server accesses the repository.
## Possible values are: unique, failovermain, failoverstandby.
## Default value is: unique.
################################################
#ebx.repository.ownership.mode=unique

## Activation key used in case of failover. The backup server must include this
## key in the HTTP request used to transfer exclusive ownership of the repository.
## The activation key must be an alphanumeric ASCII string longer than 8 characters.
#ebx.repository.ownership.activationkey=

## Specifies whether to hide heartbeat logging in DEBUG mode.
## Default value is true.
#ebx.repository.ownership.hideHeartBeatLogForDebug=true
```

59.18 Tuning the EBX repository

Some options can be set so as to optimize memory usage.

The properties are configured as follows:

```
################################################################
## Technical parameters for memory and performance tuning
################################################################
# Import commit threshold allows to specify the commit threshold
# exclusively for the archive import launched directly from Manager.
#
# For more details about the commit threshold,
# see the JavaDoc ProcedureContext.setCommitThreshold().
# Default value is 0.
#
ebx.manager.import.commit.threshold=100

# A validation messages threshold allows specifying the maximum number of
# messages to consider per constraint when performing a validation.
# This threshold is considered for each constraint defined in a data model
# and for each severity in each dataset validation report.
# When the threshold is reached by a constraint and a severity:
# - the validation of the constraint is stopped
# - an error message indicating that the threshold has been reached
# is added to the validation report.
# When set to 0 or a negative value, the threshold is not considered.
# Default value is 0.
#
ebx.validation.constraints.messages.threshold = 100


# Specifies whether the validation report should be kept in memory,
# regardless of the loading strategy of the dataspace.
# Default value is true. However, it is recommended to deactivate it
# when the repository contains a large number of open dataspaces and
# datasets.
#ebx.validation.report.keepInMemory=false
```

Voir aussi*Validation report page* *[p 389]*

59.19 Miscellaneous

Activating data workflows

This parameter specifies whether data workflows are activated. This parameter is not taken into account on the fly. The server must be restarted whenever the value changes.

```
################################################
## Workflow activation.
## Default is false.
################################################
ebx.workflow.activation = true
```

Disabling user task legacy mode

This parameter specifies whether the creation service of a user task in legacy mode should be offered in the workflow modeling. The default value is `true`.

See `UserTask.UserTaskMode.LEGACY_MODE`[API] for more information.

```
## Disables legacy work item mode(default is true)
## Specify if the creation service of user task in legacy mode must be offered
## in workflow modeling.
#ebx.manager.workflow.legacy.userTaskMode=false
```

Log procedure starts

This parameter specifies whether starts of the procedure execution are logged.

```
################################################
## Specifies whether transaction starts are logged. Default is false.
################################################
ebx.logs.logTransactionStart = true
```

Log validation starts

This parameter specifies whether starts of datasets validation are logged.

```
################################################
## Specifies whether validation starts are logged. Default is false.
################################################
ebx.logs.logValidationStart = true
```

Deployment site identification

This parameter allows specifying the email address to which technical log emails are sent.

```
################################################
## Unique Site Name
## --> used by monitoring emails and by the repository
################################################
ebx.site.name= name@domain.com
```

Dynamically reloading the main configuration

Some parameters can be dynamically reloaded, without restarting EBX. The parameter `thisfile.checks.intervalInSeconds` indicates how frequently the main configuration file is checked.

```
################################################
### Checks if this file has been updated
### If value <= 0, no more checks will be done
################################################
thisfile.checks.intervalInSeconds=1
```

In development mode, this parameter can be set to as low as one second. On production systems, where changes are expected to be less frequent, the value can be greater, or set to '0' to disable hot reloading entirely.

This property is not always supported when the module is deployed as a WAR, as it would then depend on the application server.

Declaring modules as undeployed

On application server startup, the initialization of deployed web applications / EBX modules and the initialization of the EBX repository are performed asynchronously. In order to properly initialize the EBX repository, it is necessary to compile all the data models used by at least a dataset, hence EBX will wait endlessly for referenced modules to be registered.

If a module is referenced by a data model but is not deployed (or no longer deployed), it is necessary to declare this module as undeployed to unlock the wait and continue the startup process.

> **Note**
>
> The `kernel` logging category indicates which modules are awaited.

> **Note**
>
> A module declared as undeployed cannot be registered into EBX until it is removed from the property `ebx.module.undeployedModules`.

> **Note**
>
> Any data model based on an unregistered module will have an "undeployed module" compilation error.

Voir aussi

Module registration *[p 484]*

Dynamically reloading the main configuration *[p 385]*

```
################################################
## Comma-separated list of EBX® modules declared
## as undeployed.
## If a module is expected by the EBX® repository but is
## not deployed, it must be declared in this property.
## Caution:
## if the "thisfile.checks.intervalInSeconds" property is deactivated,
## a restart is mandatory, otherwise it will be hot-reloaded.
################################################
ebx.module.undeployedModules=
```

Module public path prefix

EBX modules' public paths are declared in the 'module.xml' file of each module. A context prefix can be declared for all modules, without having to modify the 'module.xml' content, by specifying the property that follows.

This prefix will apply to any EBX module, including core, add-on and specific modules.

When proxy and / or reverse-proxy mode are used, the `ebx.servlet.http[s].path` and `ebx.servlet.http[s].ui.path` properties must take into account this module public path prefix setting. Conversely, the `ebx.externalResources.http[s].path` property must end its path just before a potential prefix.

```
################################################
ebx.servlet.useLocalUrl=true

#ebx.servlet.http.host=
#ebx.servlet.http.port=
ebx.servlet.http.path=reverse-proxy/prefix/ebx/
ebx.servlet.http.ui.path=reverse-proxy/prefix/ebx-ui/
#ebx.servlet.http.useHttpsSettings=false

#ebx.servlet.https.host=
#ebx.servlet.https.port=
ebx.servlet.https.path=reverse-proxy/prefix/ebx/
ebx.servlet.https.ui.path=reverse-proxy/prefix/ebx-ui/

################################################
## Web parameters (for external resources)
## if nothing is set, values are taken from servlet.
################################################
ebx.externalResources.useLocalUrl=true

#ebx.externalResources.http.host=
#ebx.externalResources.http.port=
```

```
ebx.externalResources.http.path=reverse-proxy/
#ebx.externalResources.http.useHttpsSettings=false

#ebx.externalResources.https.host=
#ebx.externalResources.https.port=
ebx.externalResources.https.path=reverse-proxy/

######################################################################
## EBX® Module context path prefix
##
## If defined, applies to all EBX® modules public paths declared in
## any module.xml file (core, add-on and specific).
######################################################################
ebx.module.publicPath.prefix=prefix/
```

See URLs computing [p 380] for more information.

EBX run mode

This property defines how EBX runs. Three run modes are available: *development*,*integration* and *production*.

When running in *development* mode, the development tools [p 495] are activated in EBX, some features thus become fully accessible and more technical information is displayed.

> **Note**
>
> The administrator can always access this information regardless of the mode used.

The additional features accessible when running in *development* mode include the following (non-exhaustive list):

Documentation pane	In the case of a computed value, the Java class name is displayed. A button is displayed giving access to the path to a node.
Compilation information	Module and schema compilation information is displayed in the dataset validation report.
Java bindings	The generation of Java bindings is available if the schema of the dataset mentions at least one binding.
Gnrateur de liens pour composant Web	The Gnrateur de liens pour composant Web is available on datasets and dataspaces.
Data model assistant	Data model configuration and additional options, such as Services, Business Objects and Rules, Java Bindings, Toolbars and some advanced properties.
Workflow modeling	Declare specific script tasks.
Log	The logs include additional technical information intended for the developer. For example, a warning is written to logs if a drop-down list is defined on a node which is not an enumeration in a UI Bean.
Product documentation	The product documentation is always the most complete one (i.e "advanced"), including administration and development chapters.

```
################################################
## Server Mode
## Value must be one of: development, integration, production
## Default is production.
################################################
backend.mode=integration
```

> **Note**
>
> There is no difference between the *integration* and *production* modes.

Resource filtering

This property allows the filtering of certain files and directories in the resource directory contents (resource type node, with an associated facet that indicates the directory that contains usable resources).

```
################################################
## list (separated by comma) of regexps excluding resource
## the regexp must be of type "m:[pattern]:[options]".
## the list can be void
################################################
```

```
ebx.resource.exclude=m:CVS/*:
```

Validation report page

The validation report page can display a finite number of items for each severity. This number can be tuned with this property.

```
# Defines the maximum item displayed for each severity in the validation report page.
# Default value is 100.
#ebx.validation.report.maxItemDisplayed=100
```

Voir aussi*Tuning the EBX repository* *[p 384]*

Validation report logs

This property allows to specify the number of validation messages to display in the logs when validating a dataset or a table.

```
# Defines the maximum number of messages displayed in the logs.
# Default value is 100.
# When set to 0 or a negative value, the limit is not considered.
#ebx.validation.report.maxItemDisplayedInLogs=500
```

Voir aussi*Tuning the EBX repository* *[p 384]*

CHAPITRE 60

Initialization and first-launch assistant

Deliverables can be found on TIBCO eDelivery(an account is mandatory in order to access eDelivery, please contact the support team to request one).

The TIBCO EBX Configuration Assistant helps with the initial configuration of the EBX repository. If EBX does not have a repository installed upon startup and if the automatic installation [p 370] is not enabled, the configuration assistant is launched automatically.

Before starting the configuration of the repository, make sure that EBX is correctly deployed on the application server. See Java EE deployment [p 341].

> **Note**
>
> The EBX main configuration file must also be properly configured. See TIBCO EBX main configuration file [p 369].

Ce chapitre contient les sections suivantes :

1. License key
2. Configuration steps

60.1 License key

When launching EBX, the license key page displays automatically if no valid license key is available, that is, if there is no license key entered in the EBX main configuration file, or if the current license key has expired.

The license key can be retrieved from the TIBCO eDelivery site (an account is mandatory in order to access eDelivery, please contact https://support.tibco.com to request one).

The license key is available in the `TIB_ebx_{ebx.version.public}_license_key.pdf` file.

60.2 Configuration steps

The EBX configuration assistant guides you through the following steps:

1. Validating the license agreement.
2. Configuring the repository.

3. Defining users in the default user and roles directory (if a custom directory is not defined).
4. Validating the information entered.
5. Installing the EBX repository.

Validating the license agreement

In order to proceed with the configuration, you must read and accept the product license agreement.

Configuring the repository

This page displays some of the properties defined in the EBX main configuration file. You also define several basic properties of the repository in this step.

Id of the repository (`repositoryId`)	Must uniquely identify the repository (in the scope of the enterprise). The identifier is 48 bits (6 bytes) long and is usually represented as 12 hexadecimal digits. This information is used for generating the Universally Unique Identifiers (UUIDs) of entities created in the repository, and also of transactions logged in the history. This identifier acts as the "UUID node", as specified by RFC 4122.
Repository label	Defines a user-friendly label that indicates the purpose and context of the repository.

Voir aussi*TIBCO EBX main configuration file* *[p 369]*

Defining users in the default directory

If a custom user and roles directory is not defined in the EBX main configuration file, the configuration assistant allows to define default users for the default user and roles directory.

An administrator user must be defined. You may optionally create a second user.

Voir aussi*Users and roles directory* *[p 423]*

Validating the information entered

Before proceeding with the installation of the repository, you can review the configuration of the repository and the information entered on the 'Configuration Summary' page. If you need to modify information, you can return to the previous pages using the configuration assistant **< Back** button.

Once you have verified the configuration, click the button **Install the repository >** to proceed with the installation.

Installing the EBX repository

The repository installation is performed using the provided information. When the installation is complete, you are redirected to the repository login page.

CHAPITRE 61

Deploying and registering TIBCO EBX add-ons

> **Note**
>
> Refer to the documentation of each add-on for additional installation and configuration information in conjunction with this documentation.

Ce chapitre contient les sections suivantes :

1. Deploying an add-on module
2. Registering an add-on module
3. Deleting an add-on module

61.1 Deploying an add-on module

> **Note**
>
> **Each add-on bundle version is intended to run with a specific EBX version and all its fix releases. Make sure that the EBX and add-on bundle versions are compatible, otherwise the add-on registration will abort.**

The web application deployment descriptor for the add-on module must specify that class definitions and resources from the web application are to be loaded in preference to classes from the parent and server classloaders.

For example, on WebSphere Application Server, this can be done by setting `<context-priority-classloader>true</context-priority-classloader>` in the `web-app` element of the deployment descriptor.

On WebLogic, include `<prefer-web-inf-classes>True</prefer-web-inf-classes>` in `weblogic.xml`.

See the documentation on class loading of your application server for more information.

The EBX add-on common JAR file, named `lib/ebx-addons.jar`, must be copied in the library directory shared by all web applications.

> **Note**
>
> The add-on log level can be managed in the main configuration file [p 377].

61.2 Registering an add-on module

To register a new EBX add-on in the repository:

1. Navigate to the 'Administration' area.
2. Click the down-arrow in the navigation pane and select **Technical configuration** > **Add-ons registration**.
3. On the **Registered add-ons** page, click the **+** button to create a new entry.
4. Select the add-on you are registering, and enter its license key.

   > **Note**
   >
   > If the EBX repository is under a trial license, no license key is required for the add-on. The add-on will be subject to the same trial period as the EBX repository itself.

   The license key can be retrieved from the TIBCO eDelivery site.
5. Click on **Save**.

61.3 Deleting an add-on module

To delete an add-on module from the EBX repository:

1. Navigate to the 'Administration' area.
2. Click the down-arrow in the navigation pane and select **Technical configuration** > **Add-ons registration**.
3. On the **Registered add-ons** page, tick the box corresponding to the add-on to be deleted.
4. In the 'Actions' menu, select 'Delete'.
5. Close and purge the Administration datasets related to the previously used add-on, as well as the including dataspaces.

   When an add-on is no longer deployed, a dataspace corresponding to the Administration dataset will then appear in the list of Reference children under the dataspaces. When an add-on module is no longer deployed, it is thus necessary to close/delete and purge manually all data/dataspaces related to the add-on.

Technical administration

CHAPITRE 62

Repository administration

Ce chapitre contient les sections suivantes :

1. Technical architecture
2. Auto-increments
3. Repository management
4. Monitoring management
5. Dataspaces

62.1 Technical architecture

Overview

The main principles of the TIBCO EBX technical architecture are the following:

- A Java process (JVM) that runs EBX is limited to a single EBX repository. This repository is physically persisted in a supported relational database instance [p 337], accessed through a configured data source [p 371].
- A repository cannot be shared by multiple JVMs at any given time. However, a failover architecture may be used. These aspects are detailed in the sections Single JVM per repository [p 397] and Failover with hot-standby [p 397]. Furthermore, to achieve horizontal scalability, an alternative is to deploy a distributed data delivery (D3) [p 448] environment.
- A single relational database instance can support multiple EBX repositories (used by distinct JVMs). It is then required that they specify distinct table prefixes using the property `ebx.persistence.table.prefix`.

Voir aussi

Configuring the EBX repository [p 371]

Supported databases [p 337]

Data source of the EBX repository [p 347]

Rules for the database access and user privileges

Attention

In order to guarantee the integrity of persisted master data, **it is strictly forbidden to perform direct SQL writes to the database**, except for specific use cases described in the section Accs SQL aux donnes en mode relationnel [p 271].

It is required for the database user specified by the configured data source [p 371] to have the 'create/alter' privileges on tables, indexes and sequences. This allows for automatic repository installation and upgrades [p 399].

Voir aussi

Accs SQL l'historique [p 276]

Accder une table rplique l'aide de SQL [p 283]

Accs SQL aux donnes en mode relationnel [p 271]

Data source of the EBX repository [p 347]

Single JVM per repository

A repository cannot be shared by multiple JVMs. If such a situation was to occur, it would lead to unpredictable behavior and potentially even corruption of data in the repository.

EBX performs checks to enforce this restriction. Before the repository becomes available, the repository must first acquire exclusive ownership of the relational database. After starting the repository, the JVM periodically checks that it still holds ownership of the repository.

These checks are performed by repeatedly tagging a technical table in the relational database. The shutdown command for the application server ensures that the tag on this technical table is removed. If the server shuts down unexpectedly, the tag may be left in the table. If this occurs, the server must wait several additional seconds upon restart to ensure that the table is not being updated by another live process.

Attention

To avoid an additional wait period at the next start up, it is recommended to always properly shut down the application server.

Failover with hot-standby

The exclusion mechanism described above is compatible with failover architectures, where only one server is active at any given time in an active/passive cluster. To ensure that this is the case, the main server must declare the property `ebx.repository.ownership.mode=failovermain`. The main server claims ownership of the repository database, as in the case of a single server.

A backup server can still start up, but it will not have access to the repository. It must declare the property `ebx.repository.ownership.mode=failoverstandby` to act as the backup server. Once started, the backup server is registered in the connection log. Its status can be retrieved using the Java API or through an HTTP request, as described in the section Repository status information and logs [p 398] below.

In order to activate the backup server and transfer exclusive ownership of the repository to it, a specific request must be issued by an HTTP request, or using the Java API:

- Using HTTP, the request must include the parameter `activationKeyFromStandbyMode`, and the value of this parameter must be equal to the value declared for the entry `ebx.repository.ownership.activationkey` in the EBX main configuration file. See Configuring failover [p 383].

  The format of the request URL must be:

  `http[s]://<host>[:<port>]/ebx?activationKeyFromStandbyMode={value}`
- Using the Java API, call the method `RepositoryStatus.wakeFromStandby`[API].

If the main server is still up and accessing the database, the following applies: the backup server marks the ownership table in the database, requesting a clean shutdown for the main server (yet allowing any running transactions to finish). Only after the main server has returned ownership can the backup server start using the repository.

Repository status information and logs

A log of all attempted Java process connections to the repository is available in the Administration area under 'History and logs [p 273]' > 'Repository connection log'.

The status of the repository may be retrieved using the methods in the `RepositoryStatus`[API] API.

It is also possible to get the repository status information using an HTTP request that includes the parameter **`repositoryInformationRequest`** with one of following values:

`state`	The state of the repository in terms of ownership registration. • `D`: Java process is stopped. • `O`: Java process has exclusive ownership of the database. • `S`: Java process is started in failover standby mode, but is not yet allowed to interact with the repository. • `N`: Java process has tried to take ownership of the database but failed because another process is holding it.
`heart_beat_count`	The number of times that the repository has made contact since associating with the database.
`info`	Detailed information for the end-user regarding the repository's registration status. The format of this information may be subject to modifications in the future without explicit warning.

62.2 Auto-increments

Several technical tables can be accessed in the 'Administration' area of the EBX user interface. These tables are for internal use only and their content should not be edited manually, unless removing obsolete or erroneous data. Among these technical tables are:

Auto-increments	Lists all auto-increment fields in the repository.

62.3 Repository management

Installation and upgrades

Automatic installation and upgrades

By complying with the Rules for the database access and user privileges [p 397], the repository installation or upgrade is done automatically.

Inter-database migration

EBX provides a way to export the full content of a repository to another database. The export includes all dataspaces, configuration datasets, and mapped tables. To operate this migration, the following guidelines must be respected:

- The source repository **must be shut down**: no EBX server process must be accessing it; **not strictly complying with this requirement can lead to a corrupted target repository**;
- A new EBX server process must be launched on the target repository, which must be empty. In addition to the classic Java system property `-Debx.properties`, this process must also specify `ebx.migration.source.properties`: the location of an EBX properties file specifying the source repository. (It is allowed to provide distinct table prefixes between target and source.)
- The migration process will then take place automatically. Please note, however, that this process is not transactional: should it fail halfway, it will be necessary to delete the created objects in the target database, before starting over.
- After the migration is complete, an exception will be thrown, to force restarting the EBX server process accessing the target repository.

Limitations:

- For technical reasons, migration to an Oracle database is only supported from another Oracle database.
- The names of the database objects representing the mapped tables (history, replication, relational) may have to be altered when migrated to the target database, to comply with the limitations of its database engine (maximum length, reserved words, ...). Such alterations will be logged during the migration process.
- As a consequence, the names specified for replicated tables in the data model will not be consistent with the adapted name in the database. The first recompilation of this data model will force to correct this inconsistency.

- Due to different representations of numeric types, values for `xs:decimal` types might get rounded if the target database engine offers a lesser precision than the source. For example, a value of `10000000.1234567890123456789` in Oracle will get rounded to `10000000.123456789012345679` in SQL Server.

Repository backup

A global backup of the EBX repository must be delegated to the underlying RDBMS. The database administrator must use the standard backup procedures of the underlying database.

Archives directory

Archives are stored in a sub-directory called `archives` within the `ebx.repository.directory` (see configuration [p 369]). This directory is automatically created during the first export from EBX.

Attention

If manually creating this directory, make sure that the EBX process has read-write access to it. Furthermore, the administrator is responsible for cleaning this directory, as EBX does not maintain it.

> **Note**
>
> The transfer of files between two EBX environments must be performed using tools such as FTP or simple file copies by network sharing.

Repository attributes

A repository has the following attributes:

repositoryId	Uniquely identifies a repository within the scope of the company. It is 48 bits (6 bytes) and is usually represented as 12 hexadecimal digits. This information is used for generating UUIDs (Universally Unique Identifiers) for entities created in the repository, as well as transactions logged in history tables or in the XML audit trail. This identifier acts as the 'UUID node' part, as specified by *RFC 4122*.
repository label	Provides a user-friendly label that identifies the purpose and context of the repository. For example: "Production environment".
store format	Identifies the underlying persistence system, including the current version of its structure.

62.4 Monitoring management

Monitoring and cleanup of the relational database

Some entities accumulate during the execution of EBX.

Attention

It is the *administrator's responsibility* to monitor and clean up these entities.

Monitoring and reorganization

The persistence data source of the repository must be monitored through RDBMS monitoring.

The EBX tables specified in the default semantic mode [p 267] have their content persisted in a set of generic database tables:

- The table `${ebx.persistence.table.prefix}HOM`, in which each record represents a dataspace or a snapshot (its name is `EBX_HOM` if the property `ebx.persistence.table.prefix` is unset).
- The table `${ebx.persistence.table.prefix}BLK`, where the data of EBX tables in semantic mode are segmented into blocks of at most 100 EBX records (its name is `EBX_BLK` if the property `ebx.persistence.table.prefix` is unset).
- The tables ${ebx.persistence.table.prefix}HTA, ${ebx.persistence.table.prefix}TAR and ${ebx.persistence.table.prefix}ATB, defining which blocks belong to a given EBX table in a given dataspace or snapshot.

Database statistics

The performance of requests executed by EBX requires that the database has computed up-to-date statistics on its tables. Since database engines regularly schedule statistics updates, this is usually not an issue. Yet, it could be necessary to explicitly update the statistics in cases where tables are heavily modified over a short period of time (e.g. by an import creating many records).

Impact on UI

Some UI components use statistics to adapt their behavior in order to prevent users from executing costly requests unwillingly.

For example, the combo box will not automatically search on user input if the table contains a large volume of records. This behavior may also occur if the database's statistics are not up to date, because a table may be considered as containing a large volume of records even if it is not actually the case.

Cleaning up dataspaces, snapshots, and history

A full cleanup of dataspaces, snapshots, and history from the repository involves several stages:

1. Closing unused dataspaces and snapshots to keep the cache to a minimal size.
2. Deleting dataspaces, snapshots, and history.
3. Purging the remaining entities associated with the deleted dataspaces, snapshots, and history from the repository.

Closing unused dataspaces and snapshots

In order to keep the cache and the repository to a reasonable size, it is recommended to close any dataspaces and snapshots that are no longer required. This can be done in the following ways:

- Through the user interface, in the 'Dataspaces' area.
- From the 'Dataspaces / Snapshots' table under 'Dataspaces' in the 'Administration' area, using the **Actions** menu in the workspace. The action can be used on a filtered view of the table.
- Through the Java API, using the method `Repository.closeHome`[API].
- Using the data service "close dataspace" and "close snapshot" operations. See Closing a dataspace or snapshot [p 665] for more information.

Once the dataspaces and snapshots have been closed, the data can be safely removed from the repository.

> **Note**
>
> Closed dataspaces and snapshots can be reopened in the 'Administration' area, under 'Dataspaces'.

Deleting dataspaces, snapshots, and history

Dataspaces, associated history and snapshots can be permanently deleted from the repository. However, the deletion of a dataspace does not necessarily imply the deletion of its history. The two operations are independent and can be performed at different times.

> **Note**
>
> The deletion of a dataspace, a snapshot, or of the history associated with them is recursive. The deletion operation will be performed on every descendant of the selected dataspace.

After the deletion of a dataspace or snapshot, some entities will remain until a repository-wide purge of obsolete data is performed. In particular, the complete history of a dataspace remains visible until a repository-wide purge is performed. Both steps, the deletion and the repository-wide purge, must be completed in order to totally remove the data and history. The process has been divided into two steps for performance issues. As the total clean-up of the repository can be time-intensive, this allows the purge execution to be initiated during off-peak periods on the server.

The process of deleting the history of a dataspace takes into account all history transactions recorded up until the deletion is submitted or until a date specified by the user. Any subsequent historized operations will not be included when the purge operation is executed. To delete new transactions, the history of the dataspace must be deleted again.

> **Note**
>
> It is not possible to set a deletion date in the future. The specified date will thus be ignored and the current date will be used instead.

The deletion of dataspaces, snapshots, and history can be performed in a number of different ways:

- From the 'Dataspaces/Snapshots' table under 'Dataspaces' in the 'Administration' area, using the **Actions** menu button in the workspace. The action can be used on a filtered view of the table.

- Using the Java API, and more specifically the methods `Repository.deleteHome`[API] and `RepositoryPurge.markHomeForHistoryPurge`[API].
- At the end of the data service "close dataspace" operation, using the parameters `deleteDataOnClose` and `deleteHistoryOnClose`, or at the end of a "merge dataspace" operation, using the parameters `deleteDataOnMerge` and `deleteHistoryOnMerge`.

Purging remaining entities after a dataspace, snapshot, or history deletion

Once items have been deleted, a purge can be executed to clean up remaining data from *all* deletions performed until that point. A purge can be initiated in the following ways:

- Through the user interface, by selecting in the 'Administration' area **Actions > Execute purge** in the navigation pane.
- Using the Java API, specifically the method `RepositoryPurge.purgeAll`[API].
- Using the task scheduler. See Task scheduler [p 437] for more information.

The purge process is logged in the directory `${ebx.repository.directory}/db.purge/`.

Cleaning up tables having unreadable records

Some data model evolutions can lead to unreadable data:

- A column containing null values is added to the primary key of a table.
- The type of a column has changed to a different type with no possible conversion.

In these situations, records that do not match the new table definition are no longer visible, but remain persisted in the table. (This allows retrieving records if switching back to the previous table definition). When such records are encountered, an informative error is recorded in EBX logs.

EBX provides the option to clean the records that no longer conform to the model, once the new version of the data model is stabilized. This allows recovering space in the database and getting rid of error messages. Please proceed carefully, as this operation permanently removes all unreadable records from the selected table, and cannot be undone.

Cleaning up unreadable records is done by selecting in the 'Administration' area **Actions > Clean up unreadable records** in the navigation pane.

Cleaning up other repository entities

It is the *administrator's responsibility* to monitor and regularly cleanup the following entities.

Purge

A purge can be executed to clean up the remaining data from *all* deletions, that is, deleted dataspaces, snapshots and history performed up until that point. A purge can be initiated by selecting in the 'Administration' area **Actions > Execute purge** in the navigation pane.

Task scheduler execution reports

Task scheduler execution reports are persisted in the 'executions report' table, in the 'Task scheduler' section of the 'Administration' area. Scheduled tasks constantly add to this table as they are executed. Even when an execution terminates normally, the records are not automatically deleted. It is thus recommended to delete old records regularly.

User interactions

User interactions are used by the EBX component as a reliable means for an application to initiate and get the result of a service execution. They are persisted in the *ebx-interactions* administration section. It is recommended to regularly monitor the user interactions table, as well as to clean it, if needed.

Workflow history

The workflow events are persisted in the workflow history table, in the 'Workflow' section of the 'Administration' area. Data workflows constantly add to this table as they are executed. Even when an execution terminates normally, the records are not automatically deleted. It is thus recommended to delete old records regularly.

The steps to clean history are the following

- Make sure the process executions are removed (it can be done by selecting in the 'Administration' area of Workflows **Actions > Terminate and clean this workflow** or **Actions > Clean from a date** in the navigation pane).
- Clean main processes in history (it can be done by selecting in the 'Administration' area of Workflows history **Actions > Clear from a date** or **Actions > Clean from selected workflows** in the navigation pane).
- Purge remaining entities in workflow history using 'standard EBX purge'

Voir aussi*the standard EBX purge [p 402]*

Monitoring and clean up of file system

Attention

In order to guarantee the correct operation of EBX, the disk usage and disk availability of the following directories must be supervised by the administrator, as EBX does not perform any clean up:

- **XML audit trail**: `${ebx.repository.directory}/History/`
- **Archives**: `${ebx.repository.directory}/archives/`
- **Logs**: ebx.logs.directory [p 375]
- **Temporary directory**: ebx.temp.directory [p 373]

Attention

For **XML audit trail**, if large transactions are executed with full update details activated (default setting), the required disk space can increase.

Attention

For pagination in the data services `getChanges` operation, a persistent store is used in the **Temporary directory**. Large changes may require a large amount of disk space.

Voir aussi

XML audit Trail [p 443]

Tuning the EBX repository *[p 384]*

62.5 Dataspaces

Some dataspace administrative tasks can be performed from the 'Administration' area of EBX by selecting 'Dataspaces'.

Dataspaces/snapshots

This table lists all the existing dataspaces and snapshots in the repository, whether open or closed. You can view and modify the information of dataspaces included in this table.

Voir aussi*Dataspace information* *[p 104]*

From this section, it is also possible to close open dataspaces, reopen previously closed dataspaces, as well as delete and purge open or closed dataspaces, associated history, and snapshots.

Voir aussi*Cleaning up dataspaces, snapshots, and history* *[p 401]*

Dataspace permissions

This table lists all the existing permission rules defined on all the dataspaces in the repository. You can view the permission rules and modify their information.

Voir aussi*Permissions sur un espace de donnes* *[p 106]*

Repository history

The table 'Deleted dataspaces/snapshots' lists all the dataspaces that have already been purged from the repository.

From this section, it is also possible to delete the history of purged dataspaces.

CHAPITRE 63

UI administration

TIBCO EBX comes with a full user interface called Advanced perspective [p 408] that includes all available features. The interface is fully customizable [p 411] (custom logo, colors, field size, default values, etc.) and available to built-in administrators.

Access to the advanced perspective can be restricted in order to simplify the end-user experience, through global permissions [p 407], giving the possibility to grant or restrict access to functional categories. Administrators can create simplified perspectives called recommended perspectives [p 419] for end-users, containing only the features and menus they need for their daily tasks.

Ce chapitre contient les sections suivantes :

1. Global permissions
2. Advanced perspective
3. Recommended perspectives
4. Custom views
5. User session management

63.1 Global permissions

Global permission rules can be created in EBX.

The 'Display area' property allows restricting access to areas of the user interface. To define the access rules, select 'Global permissions' in the 'Administration' area.

Profil	Indicates on which profile the rule will be applied.
Restriction d'accs	Indique si les permissions dfinies ici restreignent celles dfinies pour d'autres profils. See the Restriction policy concept [p 307] for more information.
Espaces de donnes	Dfinit les permissions pour la section Espaces de donnes.
Modles de donnes	Dfinit les permissions pour la section Modles de donnes.
Modles de workflow	Dfinit les permissions pour la section Modles de workflow.
Workflows de donnes	Dfinit les permissions pour la section Workflows de donnes.
Services de donnes	Dfinit les permissions pour la section Services de donnes. Independently, it is also possible to: • Dfinit les permissions d'accs aux services REST prdfinis sur http(s). • Dfinit les permissions d'accs au connecteur SOAP sur http(s) et JMS. • Dfinit les permissions d'accs au connecteur WSDL sur http(s).
Administration	Dfinit les permissions pour la section Administration.

> **Note**
>
> Permissions can be defined by administrators and by the dataspace or dataset owner.

63.2 Advanced perspective

The advanced perspective and its parameterization are unique. It is the parent perspective from which any new perspective [p 419] will inherit.

Children perspectives can be created from that main perspective in order to offer a customized, simplified menu to the end-users. Thanks to dataset inheritance, these simplified perspectives will receive their parameters from the advanced perspective (the root dataset). These parameters can then be overridden on the newly created simplified perspectives. Simplified perspectives can be created underneath an existing simplified perspective, thus inheriting from the parent's parameters.

Voir aussi*Inheritance [p 27]*

The advanced perspective is available by default to all end-users but access can be restricted.

Note: Administrators can always access the advanced perspective even when it is deactivated.

It is possible to configure which perspective is applied by default when users log in. This 'default perspective' is based on two criteria: 'recommended perspectives', defined by administrators and 'favorite perspectives', defined by users.

Voir aussi

Recommended perspectives *[p 419]*

Favorite perspectives *[p 19]*

Perspective creation

To create a perspective, open the 'Select an administration feature' drop-down menu and click on the + sign to create a child dataset.

Voir aussi*Creating an inheriting child dataset* *[p 124]*

User interface

Options are available in the Administration area for configuring the web interface, in the 'User interface' section.

Attention

Be careful when configuring the URL policy (deprecated) [p 410]. If the web interface configuration is invalid, it can lead to the unusability of EBX. If this occurs, use the "rescue mode" by setting `frontEnd.rescueMode.enable=true` in EBX main configuration file [p 369], and accessing the following URL in your browser as a built-in administrator user: `http://.../ebx/?onwbpID=iebx-manager-rescue`.

Session configuration

These parameters configure the user session options:

User session default locale	Default session locale
Session time-out (in seconds)	Maximum duration of user inactivity before the session is considered inactive and is terminated. A negative value indicates that the session should never timeout.

Interface configuration

Entry policy

Describes the URL to access the application.

Login URL	If the user is not authenticated, the session is forwarded to this URL.

The entry policy defines an EBX login page, replacing the default one.

If defined,

- it replaces an authentication URL that may have been defined using a specific user `Directory`[API],
- it is used to build the permalinks in the user interface,
- if the URL is full, that is, starting with `http://` or `https://`, it replaces the URL of the workflow email configuration.

URL policy (deprecated)

Describes the URL and proxy policy. Both dynamic (servlet) and static (resources) URLs can be configured.

This configuration manner is deprecated and must be replaced by URLs computing [p 380]. After configuring the EBX main configuration file, these configurations must be unset.

HTTP servlet policy	Header content of the servlet HTTP request: • if a field is not set, the default value in the environment configuration is used, • if a default value is not set, the value in the initial request is used.
HTTPS servlet policy	Header content of the servlet HTTPS request: • if a field is not set, the default value is chosen (in an environment configuration), • if a default value is not set, the value in the initial request is used.
HTTP external resources policy	Header content of the external resources URL in HTTP: • if a field is not set, the default value in the environment configuration is used, • if a default value is not set, the value in the initial request is used.
HTTPS external resources policy	Header content of the external resources URL in HTTPS: • if a field is not set, the default value in the environment configuration is used, • if a default value is not set, the value in the initial request is used.

Exit policy

Describes how the application is exited.

Normal redirection	Specifies the redirection URL used when exiting the session normally.
Error redirection	Specifies the redirection URL used when exiting the session due to an error. This feature is now deprecated and may be ignored by EBX.
Redirection restrictions	Specifies the list of authorized domains and whether HTTPS is mandatory for each domain.

Graphical interface configuration

Activation & Allowed profiles

The 'Activated' radio button allows to activate or deactivate the perspective. When deactivated, the perspective will only be made available to the administrator.

The 'Allowed profiles' feature is used to give access to the perspective to a given profile. Several profiles can be added to the list of authorized profiles by clicking on the + icon below the numbered list.

The available perspective properties are:

Activ	Indique que la perspective est visible des utilisateurs autoriss.
Profils Autoriss	La liste des profils utilisateur autoriss pour la perspective.
Appareils autoriss	La liste des appareils autoriss pour la perspective. If not specified, only "EBX Web Application" can display this perspective.
Slection par dfaut	L'lment de menu qui est slectionn par dfaut. This property is not available for the advanced perspective.

Application locking

EBX availability status:

Disponibilit	Pour maintenance, l'application peut tre ferme au public (mais toujours accessible aux administrateurs).Les paramtres dfinis sont appliqus immdiatement.
Message d'indisponibilit	Message affich aux utilisateurs quand l'accs est restreint aux administrateurs.

Security policy

EBX access security policy. These parameters only apply to new HTTP sessions.

Restriction d'accs IP	Restreindre l'accs aux adresses IP dclares (voir ci-dessous).
Description de restriction IP	Regular expression representation of IP addresses authorized to access EBX. For example, `((127\.0\.0\.1) | (192\.168\.*\.*))` grants access to the local machine and the network IP range `192.168.*.*`.
Unicit de session	Specifies whether EBX should control the uniqueness of user sessions. When set to 'Yes', if a user does not log out before closing the browser, it will not be possible for that user to log in again until the previous session has timed out.

Ergonomics and layout

EBX ergonomics parameters:

Nombre maximal de colonnes d'une table	En fonction des performances du rseau et du navigateur, ajustez le nombre maximal de colonnes d'une table afficher (dans le contenu d'un jeu de donnes). Cette proprit n'est pas prise en compte lorsqu'une vue est applique sur une table.
Largeur automatique maximale des colonnes de tables	Cette valeur dfinit une largeur automatique maximale pour chaque colonne lors de l'initialisation de la table. Ceci permet d'viter que les colonnes avec un contenu trs long (tel qu'une URL) ne prennent trop de largeur. La largeur des colonnes reste modifiable avec la souris au del de cette valeur.
Nombre maximal d'lments dvelopps d'une hirarchie	Dfinit, pour les hirarchies, la limite du nombre d'lments qui peuvent tre dvelopps par l'action "Dvelopper tout". Une valeur infrieur ou gale 0 dsactive ce paramtre.
Filtre de table slectionn par dfaut	Dfinit le filtre de table slectionn par dfaut dans la liste des filtres affichs avec la vue tabulaire. En cas de modification, les utilisateurs doivent se dconnecter et se reconnecter afin d'utiliser la nouvelle valeur.
Afficher la boîte de messages automatiquement	Defines the message severity threshold for displaying the messages pop-up.
Mode de compatibilit IE	Defines whether or not to compensate for Internet Explorer 8+ displaying EBX in compatibility mode. In order to prevent Internet Explorer browsers from using compatibility mode when displaying the repository user interface, the meta-tag *http-equiv="X-UA-Compatible" content="IE=EmulateIE8"* is added to the header of pages. However, in some local environments, this setting may conflict with existing policies, in which case this header must be omitted by setting the parameter to 'No'. The default value is 'Yes'. See Specifying Document Compatibility Modes for more information.
Formulaires : largeur des libells	La largeur des libells des champs dans les formulaires.
Formulaires : largeur des champs	La largeur des champs de saisie dans les formulaires.

Formulaires : hauteur des champs texte	La hauteur des champs de saisie de type texte dans les formulaires.
Formulaires : dition de liste	Nombre de lignes caches gnrer dans l'interface utilisateur, disponibles pour crer de nouvelles occurrences dans la liste.
Formulaires : largeur de l'diteur HTML	La largeur de l'diteur HTML dans les formulaires.
Formulaires : hauteur de l'diteur HTML	La hauteur de l'diteur HTML dans les formulaires.
Slection en liste : taille de page	Nombre maximum de lignes transmises chaque requte du composant de slection dans une liste (utilis pour slectionner les cl trangres, les numrations, etc.).
Formulaire d'enregistrement : mode de prsentation des noeuds	En fonction des performances du rseau et du navigateur, ajustez la manire d'afficher chaque noeud non terminal du formulaire d'enregistrement. En ce qui concerne le poids de la page tlcharge, le mode lien est lger, les modes dvelopp et rduit sont plus lourds. En cas de modification de cette proprit, les utilisateurs doivent se dconnecter et se reconnecter afin que la nouvelle valeur soit prise en compte.
Formulaire d'enregistrement : affichage des noeuds de slection et d'association en cration	Si activ, les noeuds de slection et d'association seront affichs dans les formulaires de cration.
Densit d'affichage	Dfinit la densit d'affichage par dfaut pour tous les utilisateurs. Si aucune densit n'a t slectionne par l'utilisateur, c'est cette valeur qui sera applique. Dans le cas contraire, le choix de l'utilisateur prvaut.
Affichage de l'avatar dans l'en-tte	Cette proprit dfinit le mode d'affichage de l'avatar dans l'en-tte. Il est notamment possible d'activer ou dsactiver l'utilisation des avatars dans l'en-tte en modifiant cette proprit. Si aucune valeur n'est dfinie, la valeur par dfaut est 'Avatar seulement'.
Affichage de l'avatar dans l'historique	Cette proprit dfinit le mode d'affichage de l'avatar dans l'historique. Il est notamment possible d'activer ou dsactiver l'utilisation des avatars dans l'historique en modifiant cette proprit. Si aucune valeur n'est dfinie, la valeur par dfaut est 'Avatar seulement'.

Affichage de l'avatar dans le workflow	Cette proprit dfinit le mode d'affichage de l'avatar dans le workflow. Il est notamment possible d'activer ou dsactiver l'utilisation des avatars dans le workflow en modifiant cette proprit. Si aucune valeur n'est dfinie, la valeur par dfaut est 'Avatar seulement'.

Default option values

Defines default values for options in the user interface.

Import/Export

CSV : jeu de caractres	Dfinit le jeu de caractres utilis par dfaut pour les imports et les exports CSV.
CSV : sparateur de champ	Dfinit le caractre sparateur utilis par dfaut pour les imports et les exports CSV.
CSV : sparateur de liste	Dfinit le caractre sparateur de liste utilis par dfaut pour les imports et les exports CSV.
Mode d'import	Spcifie le mode utilis par dfaut pour les imports.
Valeurs XML manquantes nul	Si 'Oui', quand un nœud est manquant ou vide dans le fichier import, la valeur est considre comme 'nulle' lors de la mise jour des enregistrements existants. Si 'Non', la valeur n'est pas modifie.

Colors and themes

Customizes EBX colors and themes.

URL de l'icne du site (favicon)	Le format recommand est ICO car il est compatible avec Internet Explorer.
URL du logo (SVG)	Laissez le champ vide pour utiliser l'image PNG. L'image SVG est utilise sur les navigateurs compatibles. Le systme utilisera le logo PNG si le navigateur n'est pas compatible. Si l'image PNG n'est pas renseigne, l'image GIF/JPG sera utilise. Le logo doit avoir une hauteur maximale de 40px. Si la hauteur est suprieure 40px, elle sera rogne 40px de hauteur. La largeur du logo dtermine la position des boutons du bandeau. Si l'URL est relative, prfixez-la avec "../" afin de remonter l'URL racine de l'application.
URL du logo (PNG)	L'image PNG est utilise sur les navigateurs compatibles, sinon le systme utilisera l'image GIF/JPG. Laissez ce champ et le champ du logo SVG vide pour utiliser l'image GIF/JPG. Le logo doit avoir une hauteur maximale de 40px. Si la hauteur est suprieure 40px, elle sera rogne 40px de hauteur. La largeur du logo dtermine la position des boutons du bandeau. Si l'URL est relative, prfixez-la avec "../" afin de remonter l'URL racine de l'application.
URL du logo (GIF/JPG)	L'image GIF/JPG est utilise quand les images PNG et SVG ne sont pas renseignes. Les formats recommands sont GIF et JPG. Le logo doit avoir une hauteur maximale de 40px. Si la hauteur est suprieure 40px, elle sera rogne 40px de hauteur. La largeur du logo dtermine la position des boutons du bandeau. Si l'URL est relative, prfixez-la avec "../" afin de remonter l'URL racine de l'application.
Principale	Couleur principale de l'interface utilisateur, utilise pour les slections et les surbrillances.
Bandeau	Couleur de fond du bandeau de l'interface utilisateur. Par dfaut, dfinie la mme valeur que la couleur principale.
Pastille du workflow	Couleurs de fond et de texte/bordure de la pastille du workflow (compteur de workflows utilisateur).
Boutons primaires	Couleur des boutons slectionns par dfaut. Par dfaut, dfinie la mme valeur que la couleur principale.

Texte des boutons style lien	Couleur du texte de quelques boutons qui ont un style lien (le texte n'est ni fonc ni clair, il est color). Par dfaut, dfinie la mme valeur que la couleur principale.
Bordure de l'onglet slectionn	Couleur de bordure de l'onglet slectionn. Par dfaut, dfinie la mme valeur que la couleur principale.
Vue historique de table : donnes techniques	Couleur de fond des cellules de donnes techniques dans la vue historique de table.
Vue historique de table : cration	Couleur de fond des cellules ayant l'tat 'cration' dans la vue historique de table.
Vue historique de table : suppression	Couleur de fond des cellules ayant l'tat 'suppression' dans la vue historique de table.
Vue historique de table : mise jour	Couleur de fond des cellules ayant l'tat 'mise jour' dans la vue historique de table.

Child perspective menu

An unlimited number of child perspectives can be created. Child perspectives inherit from the parameters of the 'Advanced perspective'. Some of these parameters can be overridden as detailed hereafter.

Activation & Allowed profiles

See Activation and Allowed profiles for the Advanced perspective [p 411] for more information.

> **Note**
>
> Any specific parameter set for this perspective will override the default parameters that have been set in the 'Advanced perspective' configuration.

Perspective Menu

This view displays the perspective menu. It is a hierarchical table view.

From this view, a user can create, delete or reorder menu item records.

Voir aussi*Hierarchical table view [p 27]*

Section Menu Item	This is a top level menu item. It contains other menu items.
Menu group	This is a container for other menu items.
Action Menu Item	This menu item displays a user service in the workspace area.

Menu item properties

When creating a record in the 'Perspective' Menu, the available perspective properties are:

Type	Le type de l'lment de menu **Voir aussi***Menu item types* *[p 417]*
Parent	Le parent de l'lment de menu. This property is not available for section menu items.
Libell	Le libell de l'lment de menu. The label is optional for action menu items. If not specified, the label will be dynamically generated by EBX when the menu item is displayed.
Appareils autoriss	La liste des appareils autoriss pour cet lment. If not specified, all devices can display this menu item. Currently only two devices are supported:"EBX Web Application" and "EBX GO".
Icne	L'icne pour l'lment de menu. Icon can be either "standard" (provided by EBX) or an image, specified by a URL, that can be hosted on any web server. Icons size should be 16x16 pixels. This property is not available for section menu items.
Sparateur haut	Indique que la section lment de menu a un sparateur haut. This property is only available for section menu items.
Action	The user service to execute when the user clicks on the menu item. **Voir aussi***User interface services* *[p 587]* If an end-user is allowed to view the perspective but not to execute the user service, an "access denied" message will be displayed when the user clicks on the menu item. This property is only available for action menu items.
Selection aprs fermeture	L'lment de menu qui sera slectionn quand le service se terminera. Built-in services use this property when the user clicks on the 'Close' button.

This property is only available for action menu items.

Ergonomics and layout

See Ergonomics and layout for the Advanced perspective [p 413] for more information.

> **Note**
>
> Any specific parameter set for this perspective will override the default parameters that have been set in the 'Advanced perspective' configuration.

Colors and themes

See Colors and themes for the Advanced perspective [p 416] for more information.

> **Note**
>
> Any specific parameter set for this perspective will override the default parameters that have been set in the 'Advanced perspective' configuration.

63.3 Recommended perspectives

It is possible for a perspective administrator to configure recommended perspectives dedicated to a specific audience. These recommended perspectives are a way to choose which perspective is applied by default when a user logs in, based on their role.

However, users always have the possibility to switch between the various perspectives that are available to them and to set one as their favorite. See Perspectives favorites [p 19] for more information.

To configure recommended perspectives, go to *User interface* > *Recommended perspectives* > *Manage recommended perspectives*.

Managing recommended perspectives

The main screen shows an ordered list of records associating a profile with a perspective. Note that the order here is important since a user can match more than one record (see Resolution [p 419] for more information).

- To add an entry, use the 'Create' action.
- To edit an entry, first select it in the list by clicking on it, then click on the 'Edit' action, or simply double-click on it.
- To remove an entry, first select it in the list, then click on the 'Delete' action.
- To move an entry, first select it in the list, then use the actions in the toolbar to the right of the list.

Resolution

When a user logs in, the following algorithm determines which perspective is selected by default:

```
// 1) favorite perspective
IF the user has a favorite perspective
AND this perspective is active
AND the user is authorized for this perspective
    SELECT this perspective
    DONE

// 2) recommended perspective
FOR EACH association in the recommended perspectives list, in the declared order
    IF the user is in the declared profile
```

```
    AND the associated perspective is active
    AND the user is authorized for the associated perspective
        SELECT this perspective
        DONE

// 3) advanced perspective
IF the advanced perspective is active
AND the user is authorized for this perspective
    SELECT this perspective
    DONE

// 4) any perspective
SELECT any active perspective for which the user is authorized
DONE
```

63.4 Custom views

Users can create and manage custom views directly from the 'View' menu on tables. This administration section is the central point to manage these custom views.

Vues

This table contains all custom views defined on any table. Only a subset of fields is editable:

Documentation	Libells et descriptions localiss.
Propritaire	Profil propritaire (auteur) de cette spcification de vue.
Groupe de la vue	Groupe d'appartenance de cette vue.
Partager avec	Autres profils pouvant utiliser cette vue depuis le menu 'Vue'.

Permissions des vues

This table allows to manage permissions relative to custom views, by data model and profile. The following permissions can be configured (the default value is applied when no permission is set for a given user):

Permission	Description	Default value
Recommander des vues	Autorise l'utilisateur grer les vues recommandes.	If the user is the dataset owner, the default value is 'Yes', otherwise it is 'No'.
Grer des vues	Dfinit les vues que l'utilisateur peut modifier et supprimer.	If the user is a built-in administrator, the default value is 'Owned + shared', otherwise it is 'Owned'.
Partager des vues	Dfinit les vues pour lesquelles l'utilisateur peut modifier le champ 'Partager avec'.	If the user is a built-in administrator, the default value is 'Owned + shared', else if the user is the dataset owner, it is 'Owned', otherwise it is 'None'.
Publier des vues	Autorise l'utilisateur publier des vues afin de les rendre accessibles tous les autres utilisateurs grce aux composants web, tches utilisateur du workflow, ou services de donnes.	If the user is a built-in administrator, the default value is 'Yes', otherwise it is 'No'.

63.5 **User session management**

This tool lists all user sessions and allows terminating active sessions when necessary.

For example: it is possible to invalidate and terminate all currently open and active sessions for maintenance purposes. The access to the user interface can be temporarily closed, with an unavailability message being displayed, through Application locking [p 412]. After active sessions are terminated, users will not be able to reconnect and will see the unavailability message. The maintenance operation can then be performed.

CHAPITRE **64**

Users and roles directory

Ce chapitre contient les sections suivantes :

1. Overview
2. Concepts
3. Default directory
4. Custom directory

64.1 Overview

TIBCO EBX uses a directory for user authentication and user role definition.

A default directory is provided and integrated into the EBX repository; the 'Directory' administration section allows defining which users can connect and what their roles are.

It is also possible to integrate another type of enterprise directory.

Voir aussi

Configuring the user and roles directory *[p 373]*

Custom directory *[p 426]*

64.2 Concepts

In EBX, a user can be a member of several roles, and a role can be shared by several users. Moreover, a role can be included into another role. The generic term *profile* is used to describe either a user or a role.

In addition to the directory-defined roles, EBX provides the following *built-in roles*:

Role	Definition
Profile.ADMINISTRATOR	Built-in Administrator role. Allows performing general administrative tasks.
Profile.READ_ONLY	Built-in read-only role. A user associated with the read-only role can only view the EBX repository, and has no right to perform modifications in the repository.
Profile.OWNER	Dynamic built-in owner role. This role is checked dynamically depending on the current element. It is only activated if the user belongs to the profile defined as owner of the current element.
Profile.EVERYONE	All users belong to this role.

Information related to profiles is primarily defined in the directory.

Attention

Associations between users and the built-in roles *OWNER* and *EVERYONE* are managed automatically by EBX, and thus must not be modified through the directory.

User permissions are managed separately from the directory. See Permissions [p 299].

Voir aussi

profil [p 23]

rle [p 24]

utilisateur [p 23]

administrateur [p 24]

annuaire des utilisateurs et des rles [p 24]

Policy

These properties configure the policies of the user and roles directory, for example, whether or not users can edit their own profiles.

Users

This table lists all the users defined in the internal directory. New users can be added from there.

Roles

This table lists all the users defined in the internal directory. New roles can be created in this table.

64.3 Default directory

Directory content

The default directory is represented by the dataset 'Directory', in the 'Administration' area.

This dataset contains tables for users and roles, as well as users' roles table, roles' inclusions table and salutations table.

> **Note**
>
> If a role inclusion cycle is detected, the role inclusion is ignored at the permission resolution. Refresh and check the directory validation report for cycle detection.

> **Note**
>
> Users' roles, roles' inclusions and salutations tables are hidden by default [p 568].

Depending on the policies defined, users can modify information related to their own accounts, regardless of the permissions defined on the directory dataset.

> **Note**
>
> It is not possible to delete or duplicate the default directory.

Password recovery procedure

In the default directory, passwords are encrypted (by default with a SHA256-like algorithm), and stored in this state. Consequently, it is impossible to retrieve lost passwords. A new password must be generated and sent to the user.

There are two options for this procedure:

1. A notification email is sent to the administrator, the administrator manually changes the password and sends the new password to the user.
2. A procedure automatically generates a new password and sends it to the user.

By default, the first option is used. To activate the second option, specify the property `ebx.password.remind.auto=true` in the TIBCO EBX main configuration file [p 369].

> **Note**
>
> For security reasons, the password recovery procedure is not available for administrator profiles. If required, use the administrator recovery procedure instead.

Administrator recovery procedure

If all the 'login/password' credentials of the administrators are lost, a special procedure must be followed. A specific directory class redefines an administrator user with login 'admin' and password 'admin'.

To activate this procedure:

- Specify the following property in the TIBCO EBX main configuration file [p 369]:

  ```
  ebx.directory.factory=
  com.orchestranetworks.service.directory.DirectoryDefaultRecoverFactory
  ```

- Start EBX and wait until the procedure completes.
- Reset the '`ebx.directory.factory`' property.
- Restart EBX and connect using the 'admin' account.

> **Note**
>
> While the '`ebx.directory.factory`' property is set for the recovery procedure, authentication of users will be denied.

64.4 Custom directory

As an alternative to the default directory, it is possible to integrate a specific company directory. For example, an LDAP instance, a relational database or a specific directory model instantiated into EBX.

Voir aussi *`DirectoryFactory`*[API]

CHAPITRE 65

Data model administration

Ce chapitre contient les sections suivantes :

1. Administrating publications and versions
2. Migration of previous data models in the repository
3. Schema evolutions

65.1 Administrating publications and versions

Technical data related to data model publications and versions can be accessed in the *Administration* section by an administrator.

Data Modeling contains the following two tables:

- *Publications*. Stores the publications available in the repository.
- *Versions*. Stores the versions of the data models available in the repository.

These tables are read-only but it is however possible to delete manually a publication or a version.

Important: If a publication or a version is deleted, then the content of associated datasets will become unavailable. So this technical data must be deleted with caution.

It is possible to spread this technical data to other TIBCO EBX repositories exporting an archive from an EBX repository and importing it to another one. It may be useful for propagating the evolutions of data models to other repositories.

65.2 Migration of previous data models in the repository

In versions before *5.2.0*, published data models not depending on a module were generated in the file system directory `${ebx.repository.directory}/schemas/`, with the name of the data model (*product.xsd* for example if the data model is named *Product*). Since the *5.2.0* version, this kind of data model is now fully managed within EBX through *Publications*. That is, republishing an existing data model migrates it as a *Publication* and redirects linked datasets to the new embedded data model. The previous XML Schema Document located in `${ebx.repository.directory}/schemas/` is renamed and suffixed with *toDelete*, meaning that the document is no longer used and can be safely deleted.

65.3 Schema evolutions

It is crucial to evaluate the impact of data model changes on the administration side. The following points are to be considered:

Impacts on data persistence

Administration tasks can be related to the database cleanup after a modification of the models. The following links describe how the evolutions of data models are managed at the persistence level: Cleaning up tables having unreadable records [p 403] and Purging master tables in the database [p 431].

Impacts on side features

Some components rely heavily on the data models and can be impacted by their evolutions. Some examples are: the user interface, the WSDL documents, existing archives, etc.

The 'Administration' section offers the possibility to manage some of these components (such as the views), whereas other components fall out of the administrator's scope, such as archives, WSDL files, etc.

CHAPITRE **66**

Database mapping administration

Ce chapitre contient les sections suivantes :

1. Overview
2. Renaming columns in the database
3. Purging columns in the database
4. Renaming master tables in the database
5. Renaming auxiliary tables in the database
6. Purging master tables in the database

66.1 Overview

Information and services relative to database mapping can be found in the *Administration* area.

Voir aussi

Mapped modes *[p 265]*

`DatabaseMapping`[API]

66.2 Renaming columns in the database

This feature is available on the 'Columns' table records, under the 'Actions' menu. It allows renaming a column in the database.

The administrator can specify the name of each column of the data model in the database for mapped modes.

Once the service is selected on a record, a summary screen displays information regarding the selected column and the administrator is prompted to enter a new name for the column in the database.

> **Note**
>
> It is required that the new identifier begins with a letter.
>
> Besides, the new name must be a valid column identifier, which depends on the naming rules of the underlying RDBMS.

Voir aussi `DatabaseMapping`[API]

66.3 Purging columns in the database

This feature is available on the 'Columns' table records, under the 'Actions' menu. It allows purging columns in mapped structures.

A column can be purged if it has been disabled for mapped modes.

A column is disabled for mapped modes when:

- the corresponding field has been removed from the data model, or
- the corresponding field has been changed in the data model, in a way that is not compatible (for example: its data type has been modified), or
- the defined mapped modes have been disabled locally on the corresponding fields, using the elements `osd:history` and `osd:replication`.

Voir aussi

Disabling history on a specific field or group *[p 274]*

Disabling replication on a specific field or group *[p 283]*

Note that this behavior will change for aggregated lists:

- when deactivating a complex aggregated list, its inner fields will still be in the `LIVING` state, whereas the list node is disabled. As lists are considered as auxiliary tables in the mapping system, this information can be checked in the 'Tables' table,
- on the other hand, when the deactivation is just for inner nodes of the list, then the list will remain `LIVING`, while its children will be `DISABLED IN MODEL`.

A column can be purged only if its own state is `DISABLED IN MODEL`, or if it is an inner field of a `DISABLED IN MODEL` list.

66.4 Renaming master tables in the database

This feature allows renaming master tables for both relational and history modes in the database. However, it is not available for replicated tables since their names are specified in the data model.

Both features are available on the 'Tables' table records, under the 'Actions' menu.

Master tables are database tables used for persisting the tables of the data model.

The administrator can specify in the database the name of each master table corresponding to a table of the data model.

Once the service is selected on a record, a summary screen displays information regarding the selected master table and the administrator is prompted to enter a new name for the master table in the database.

> **Note**
>
> It is required that the new identifier begins with a letter and with the repository prefix.
>
> For history tables, it is also required that the repository prefix is followed by the history tables prefix.
>
> For relational tables, it is also required that the repository prefix is followed by the relational tables prefix.
>
> Besides, the new name must be a valid table identifier, which depends on the naming rules of the underlying RDBMS.

66.5 Renaming auxiliary tables in the database

This feature allows renaming history auxiliary tables in the database. This feature is neither available for replicated tables since their names are specified in the data model, nor for the relational mode, as aggregated lists are never supported in this mode.

This feature is available on the 'Tables' table records, under the 'Actions' menu.

Auxiliary tables are database tables used for persisting aggregated lists.

The administrator can specify in the database the name of each auxiliary table corresponding to an aggregated list of the data model.

Once the service is selected on a record, a summary screen displays information regarding the selected auxiliary table and the administrator is prompted to enter a new name for the auxiliary table in the database.

> **Note**
>
> It is required that the new identifier begins with a letter.
>
> It is required that the new identifier begins with the repository prefix.
>
> It is also required that the repository prefix is followed by the history tables prefix.
>
> Besides, the new name must be a valid table identifier, which depends on the naming rules of the underlying RDBMS.

66.6 Purging master tables in the database

This feature allows purging history and relational tables in the database if they are no longer used.

It is available on the 'Tables' table records, under the 'Actions' menu, and is only available for master tables. This feature only applies to master tables. When a master table is purged, all its auxiliary tables are purged as well.

A mapped table can be purged in the database only if it has been disabled for the corresponding mapped mode.

To disable the mapped mode for a table, follow the procedure hereafter.

For history mode:

- Deactivate historization of the table in the data model, or
- Remove the table from the data model

For relational mode:

- Remove the table from the data model, or
- Deactivate the relational mode on the data model. As data models in semantic mode cannot be used for relational dataspaces, it is thus necessary to create a dataset on a semantic dataspace using this modified data model. TIBCO EBX will then detect that the relational mode was previously used, and will therefore propose the relational table database resources for purge.

CHAPITRE 67

Workflow management

Ce chapitre contient les sections suivantes :

1. Workflows
2. Interactions
3. Workflow history

67.1 Workflows

To define general parameters for the execution of data workflows, the management of workflow publications, or to oversee data workflows in progress, navigate to the 'Administration' area. Click on the down arrow in the navigation pane and select *Workflow management* > *Workflows*.

> **Note**
>
> In cases where unexpected inconsistencies arise in the workflow execution technical tables, data workflows may encounter errors. It may then be necessary to run the operation 'Clean up inconsistencies in workflow execution tables' from the 'Actions' menu in the navigation pane under *Administration* > *Workflow Management* > *Workflows*.

Execution of workflows

Various tables can be used to manage the data workflows that are currently in progress. These tables are accessible in *Workflow management* > *Workflows* in the navigation pane.

Voir aussi*Administration de workflows de donnes* *[p 193]*

Workflows table

The 'Workflows' table contains instances of all data workflows in the repository, including those invoked as sub-workflows. A data workflow is a particular execution instance of a workflow model publication. This table provides access to the data context variables for all data workflows. It can be used to access the status of advancement of the data workflow in terms of current variable values, and in case of a data workflow suspension, to modify the variable values.

From the 'Actions' menu of the 'Workflows' table, it is possible to clear the completed data workflows that are older than a given date, by selecting the 'Clean from a date' service. This service automatically ignores the active data workflows.

Tokens table

The 'Tokens' table allows managing the progress of data workflows. Each token marks the current step being executed in a running data workflow, as well as the current state of the data workflow.

Voir aussi*token* *[p 31]*

Work items table

The 'Work items' table contains all the work items associated with user tasks that currently exist. If necessary, you can manually allocate a work item to a user from this table in the case of a blockage in a data workflow. It is preferable, however, to use the buttons in the workspace of the 'Data workflows' area whenever possible to allocate, reallocate, and deallocate work items.

Voir aussi*work item* *[p 31]*

Waiting workflows table

The 'Waiting workflows' table contains all the workflows waiting for an event. If needed, a service is available to clean this table: this service deletes all lines associated with a deleted workflow.

Voir aussi*tche d'attente* *[p 30]*

Comment table

The 'Comments' table contains the user's comments for main workflows and their sub-workflows.

Workflow publications

The 'Workflow publications' table is a technical table that contains all the workflow model publications of the repository. This table associates published workflow models with their snapshots. It is not recommended to directly modify this table, but rather to use the actions available in the workflow modeling area to make changes to publications.

Configuration

Email configuration

In order for email notifications to be sent during the data workflow execution, the following settings must be configured under 'Email configuration':

- The Dfinition de l'URL field is used to build links and value mail variables in the workflow.
- The 'From email' field must be completed with the email address that will be used to send email notifications.

Interface customization

Modeling default values

The default value for some properties can be customized in this section.

The administrator has the possibility to define the default values to be used when a new workflow model or workflow step is created in the 'Workflow Modeling' section.

Work items views

Specific columns are available in the inbox and in the monitoring work items tables, in the 'Data workflows' section.

10 specific columns are available. For each specific column, a customized label can be defined.

Priorities configuration

The property 'Default priority' defines how data workflows and their work items across the repository display if they have no priority level. For example, if this property is set to the value 'Normal', any workflow and work item with no priority will appear to have the 'Normal' priority.

The 'priorities' table defines all priority levels available to data workflows in the repository. As many integer priority levels as needed can be added, along with their labels, which will appear when users hover over the priority icon in the work item tables. The icons that correspond to each priority level can also be selected, either from the set provided by TIBCO EBX, or by specifying a URL to an icon image file.

Temporal tasks

Under 'Temporal tasks', the polling interval for time-dependent tasks in the workflow can be set, such as deadlines and reminders. If no interval value is set, the 'in progress' steps are checked every hour.

Workflow inbox counter configuration

The workflow inbox counter is refreshed asynchronously, even if the end-user does not launch any action. To adjust it, two parameters need to be set:

Dure d'expiration du cache (secondes)	Dure d'expiration (en secondes) avant une nouvelle mise jour du cache de la boîte de rception. noter que ce paramtre peut avoir des consquences sur la charge CPU et les performances pour de gros volumes de donnes car le temps de calcul peut tre assez coteux. Si aucune valeur n'est renseigne, la valeur par dfaut est 600.
Priodicit de rafraîchissement de l'interface utilisateur (secondes)	Dure de rafraîchissement (en secondes) entre deux mises jour du compteur de la boîte de rception au niveau de l'interface graphique. noter que ce rafraîchissement concerne tous les compteurs de boîtes de rception contenus dans l'interface ; savoir l'en-tte et le compteur de la boîte de rception de la perspective avance. Si aucune valeur n'est renseigne, la valeur par dfaut est 5. Si la valeur renseigne est 0 (ou ngative), le rafraîchissement est dsactiv. Aussi, la modification de cette valeur ne sera effective qu'aprs reconnexion de l'utilisateur.

Also, please note that some actions can force the inbox counter to refresh:

- access on **Data workflows**
- access on any subdivision of the **Data workflows section**
- accept or reject a work item

- launch a workflow

These parameters are accessible in *Workflow management* > *Workflows* > *Configuration* > *Temporal tasks* in the navigation pane.

67.2 Interactions

To manage workflow interactions, navigate to the Administration area. Click the down arrow in the navigation pane and select the entry *Workflow management* > *Interactions*.

An *interaction* is generated automatically for every work item that is created. It is a local data context of a work item and is accessible from an EBX session. When a work item is executed, the user performs the assigned actions based upon its interaction, independently of the workflow engine. User tasks define mappings for their input and output parameters to link interactions with the overall data contexts of data workflows.

Interactions can be useful for monitoring the current parameters of work items. For example, an interaction can be updated manually by a trigger or a user service.

67.3 Workflow history

To view the history data workflow execution, browse the 'Administration' area. Click on the down arrow in the navigation pane and select *Workflow management* > *Workflow history*.

The 'Workflows' table contains all actions that have been performed during the execution of workflows.

This data can be viewed graphically or textually. It is especially useful to view the states of various objects related to workflows at a given moment. This includes actions on work items, variables in the data context, as well as tokens. In case of an error, a technical log is available.

Clean history

From the 'Actions' menu of the 'Workflows' table, the history of completed data workflows older than a given date can be cleared by selecting the 'Clear from a date' service.

Only the history of workflows that have been previously cleaned (e.g. their execution data deleted) is cleared. This service automatically ignores the history associated with existing workflows. It is necessary to clear data workflows before clearing the associated history, by using the dedicated service 'Clear from a date' from the 'Workflows' table. Also, a scheduled 'Clear from a date' can be used with the built-in scheduled task *SchedulerPurgeWorkflowMainHistory*.

Please note that only main processes are cleaned. In order to remove sub-processes and all related data, it will be necessary to run a 'standard EBX purge'.

Voir aussi*How to clean workflow history* *[p 404]*

> **Note**
>
> An API is available to fetch the history of a workflow. Direct access to the underlying workflow history SQL tables is not supported. See **WorkflowEngine.getProcessInstanceHistory** `WorkflowEngine.getProcessInstanceHistory`[API].

CHAPITRE 68

Task scheduler

Ce chapitre contient les sections suivantes :

1. Overview
2. Configuration from EBX
3. Cron expression
4. Task definition
5. Task configuration

68.1 Overview

TIBCO EBX offers the ability to schedule programmatic tasks.

> **Note**
>
> In order to avoid conflicts and deadlocks, tasks are scheduled in a single queue.

68.2 Configuration from EBX

The declaration of schedules and tasks is done by selecting 'Task scheduler' in the 'Administration' area.

- **Schedules**: defines scheduling using "cron expressions".
- **Tasks**: configures tasks, including parametrizing task instances and user profiles for their execution.
- **Scheduled tasks**: current schedule, including task scheduling activation/deactivation.
- **Execution reports**: reports of each scheduled task run that appear immediately after the task is triggered. The reports include actions to interrupt, pause, or resume running tasks, when made available by the task definition.

68.3 Cron expression

(An extract of the Quartz Scheduler documentation)

The task scheduler uses "cron expressions", which can create firing schedules such as: "At 8:00am every Monday through Friday" or "At 1:30am every last Friday of the month".

Format

A cron expression is a string comprised of 6 or 7 fields separated by a white space. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field. The fields are as follows:

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	Yes	0-59	, - * /
Minutes	Yes	0-59	, - * /
Hours	Yes	0-23	, - * /
Day of month	Yes	0-31	, - * ? / L W
Month	Yes	1-12 or JAN-DEC	, - * /
Day of week	Yes	1-7 or SUN-SAT	, - * ? / L #
Year	No	empty, 1970-2099	, - * /

A cron expression can be as simple as this: "**0 * * * * ?**",

or more complex, like this: "**0/5 14,18,3-39,52 * ? JAN,MAR,SEP MON-FRI 2002-2010**".

> **Note**
>
> The legal characters and the names of months and days of the week are not case sensitive. MON is the same as mon.

Special characters

A cron expression is a string comprised of 6 or 7 fields separated by a white space. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field. The fields are as follows:

- ***** ("all values") - used to select all values within a field. For example, "*" in the Minutes field means "every minute".
- **?** ("no specific value") - useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want my trigger to fire on a particular day of the month (say, the 10th), but don't care what day of the week that happens to be, I would put "10" in the day-of-month field, and "?" in the day-of-week field. See the examples below for clarification.
- **-** - used to specify ranges. For example, "10-12" in the hour field means "the hours 10, 11 and 12".
- **,** - used to specify additional values. For example, "MON,WED,FRI" in the day-of-week field means "the days Monday, Wednesday, and Friday".
- **/** - used to specify increments. For example, "0/15" in the seconds field means "the seconds 0, 15, 30, and 45". And "5/15" in the seconds field means "the seconds 5, 20, 35, and 50". You can also

specify '/' after the '' **character - in this case** '' is equivalent to having '0' before the '/'. '1/3' in the day-of-month field means "fire every 3 days starting on the first day of the month".

- **L** ("last") - has different meaning in each of the two fields in which it is allowed. For example, the value "L" in the day-of-month field means "the last day of the month" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "the last xxx day of the month" - for example "6L" means "the last friday of the month". When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing results.
- **W** ("weekday") - used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify "15W" as the value for the day-of-month field, the meaning is: "the nearest weekday to the 15th of the month". So if the 15th is a Saturday, the trigger will fire on Friday the 14th. If the 15th is a Sunday, the trigger will fire on Monday the 16th. If the 15th is a Tuesday, then it will fire on Tuesday the 15th. However if you specify "1W" as the value for day-of-month, and the 1st is a Saturday, the trigger will fire on Monday the 3rd, as it will not 'jump' over the boundary of a month's days. The 'W' character can only be specified when the day-of-month is a single day, not a range or list of days.

  > **Note**
  >
  > The 'L' and 'W' characters can also be combined in the day-of-month field to yield 'LW', which translates to "last weekday of the month".

- # - used to specify "the nth" day-of-week day of the month. For example, the value of "6#3" in the day-of-week field means "the third Friday of the month" (day 6 = Friday and "#3" = the 3rd one in the month). Other examples: "2#1" = the first Monday of the month and "4#5" = the fifth Wednesday of the month. Note that if you specify "#5" and there is not 5 of the given day-of-week in the month, then no firing will occur that month.

Examples

Expression	Meaning
0 0 12 * * ?	Fire at 12pm (noon) every day.
0 15 10 ? * *	Fire at 10:15am every day.
0 15 10 * * ?	Fire at 10:15am every day.
0 15 10 * * ? *	Fire at 10:15am every day.
0 15 10 * * ? 2005	Fire at 10:15am every day during the year 2005.
0 * 14 * * ?	Fire every minute starting at 2pm and ending at 2:59pm, every day.
0 0/5 14 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, every day.
0 0/5 14,18 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, AND fire every 5 minutes starting at 6pm and ending at 6:55pm, every day.
0 0-5 14 * * ?	Fire every minute starting at 2pm and ending at 2:05pm, every day.
0 10,44 14 ? 3 WED	Fire at 2:10pm and at 2:44pm every Wednesday in the month of March.
0 15 10 ? * MON-FRI	Fire at 10:15am every Monday, Tuesday, Wednesday, Thursday and Friday.
0 15 10 15 * ?	Fire at 10:15am on the 15th day of every month.
0 15 10 L * ?	Fire at 10:15am on the last day of every month.
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month.
0 15 10 ? * 6L 2002-2005	Fire at 10:15am on every last friday of every month during the years 2002, 2003, 2004 and 2005.
0 15 10 ? * 6#3	Fire at 10:15am on the third Friday of every month.
0 0 12 1/5 * ?	Fire at 12pm (noon) every 5 days every month, starting on the first day of the month.
0 11 11 11 11 ?	Fire every November 11th at 11:11am.

> **Note**
>
> Pay attention to the effects of '?' and '*' in the day-of-week and day-of-month fields.
>
> Support for specifying both a day-of-week and a day-of-month value is not complete (you must currently use the '?' character in one of these fields).
>
> Be careful when setting fire times between the hours of the morning when "daylight savings" changes occur in your locale (for US locales, this would typically be the hour before and after 2:00 AM - because the time shift can cause a skip or a repeat depending on whether the time moves back or jumps forward.

68.4 Task definition

EBX scheduler comes with some predefined tasks.

Custom scheduled tasks can be added by the means of **scheduler** `Package` `com.orchestranetworks.scheduler`[API] Java API.

The declaration of schedules and tasks is done by selecting 'Task scheduler' in the 'Administration' area.

68.5 Task configuration

A user must be associated with a task definition; this user will be used to generate the **session** `Session`[API] that will run the task.

> **Note**
>
> The user will not be authenticated, and no password is required. As a consequence, a user with no password set in the directory can only be used to run scheduled tasks.

A custom task can be parameterized by means of a JavaBean specification (getter and setter).

Supported parameter types are:

- java.lang.boolean
- java.lang.int
- java.lang.Boolean
- java.lang.Integer
- java.math.BigDecimal
- java.lang.String
- java.lang.Date
- java.net.URI
- java.net.URL

Parameter values are set in XML format.

CHAPITRE 69

Audit trail

Ce chapitre contient les sections suivantes :

1. Overview
2. Update details and disk management
3. File organization

69.1 Overview

XML audit trail is a feature that allows logging updates to XML files. An alternative history feature is also available to record table updates in the relational database; see Historique [p 273].

Any persistent updates performed in the TIBCO EBX repository are logged to an audit trail XML file. Procedure executions are also logged, even if they do not perform any updates, as procedures are always considered to be transactions. The following information is logged:

- Transaction type, such as dataset creation, record modification, record deletion, specific procedure, etc.
- Dataspace or snapshot on which the transaction is executed.
- Transaction source. If the action was initiated by EBX, this source is described by the user identity, HTTP session identifier and client IP address. If the action was initiated programmatically, only the user's identity is logged.
- Optional "trackingInfo" value regarding the session
- Transaction date and time (in milliseconds);
- Transaction UUID (conform to the Leach-Salz variant, version 1);
- Error information; if the transaction has failed.
- Details of the updates performed. If there are updates and if history detail is activated, see next section.

69.2 Update details and disk management

The audit trail is able to describe all updates made in the EBX repository, at the finest level. Thus, the XML files can be quite large and the audit trail directory must be carefully supervised. The following should be taken into account:

1. If an archive import is executed in non-interactive mode (without a change set), the audit trail does not detail the updates; it only specifies the archive that has been imported. In this case, if it is important to keep a fine trace of the import-replace, the archive itself must be preserved.
2. If an archive import is executed in interactive mode (with a change set), or if a dataspace is merged to its parent, the resulting log size will nearly triple the unzipped size of the archive. Furthermore, for consistency concerns, each transaction is logged to a temporary file (in the audit trail directory) before being moved to the main file. Therefore, EBX requires *at least six times the unzipped size of the largest archive that may be imported*.
3. In the context of a custom procedure that performs many updates not requiring auditing, it is possible for the developer to disable the detailed history using the method `ProcedureContext.setHistoryActivation`[API].

Voir aussi*EBX monitoring [p 404]*

69.3 File organization

All audit trail files are stored in the directory `${ebx.repository.directory}/History`.

"Closed" audit files

Each file is named as follows:

`<yyyy-mm-dd>-part<nn>.xml`

where `<yyyy-mm-dd>` is the file date and `<nn>` is the file index for the current day.

Writing to current audit files

When an audit file is being written, the XML structure implies working in an "open mode". The XML elements of the modifications are added to a text file named:

`<yyyy-mm-dd>-part<nn>Content.txt`

The standard XML format is still available in an XML file that references the text file. This file is named:

`<yyyy-mm-dd>-part<nn>Ref.xml`

These two files are then re-aggregated in a "closed" XML file when the repository has been cleanly shut down, or if EBX is restarted.

Example of an audit directory

```
2004-04-05-part00.xml
2004-04-05-part01.xml
2004-04-06-part00.xml
2004-04-06-part01.xml
2004-04-06-part02.xml
2004-04-06-part03.xml
2004-04-07-part00.xml
2004-04-10-part00.xml
2004-04-11-part00Content.txt
2004-04-11-part00Ref.xml
```

CHAPITRE 70

Other

Ce chapitre contient les sections suivantes :

1. Lineage
2. Event broker

70.1 Lineage

To administer lineage, three tables are accessible:

- **Authorized profiles**: Profiles must be added to this table to be used for data lineage WSDL generation.
- **History**: Lists the general data lineage WSDLs and their configuration.
- **JMS location**: Lists the JMS URL locations.

70.2 Event broker

Overview

TIBCO EBX offers the ability to receive notifications and information related to specific events using the event broker feature. This feature consists in sending notifications related to EBX core events to the subscriber according to their chosen topics.

Terminology

Event broker	Notification component for loosely-coupled event handling. Consists of dispatching fired events from EBX core to concerned subscribers. The event broker is mainly used for monitoring and statistical purposes.
Topic	Corresponds to the EBX event type that contains messages. The number of subscribers registered to a topic is unlimited.
Subscriber	Client implementation in the modules that receive the events related to the subscribed topic(s).

Topics

Dataspace and snapshot	Corresponds to operations in the dataspace and in the snapshot, such as: create, close, reopen, delete, archive export and archive import (only for dataspace merge).
Repository	Corresponds to operations in the repository, such as: start-up and purge.
User session	Corresponds to the operations related to user authentication, such as: login and logout.

Administration

The management console is located under 'Event broker' in the 'Administration' area. It contains three tables: 'Topics', 'Subscribers' and 'Subscriptions'.

All content is read-only, except for the following operations:

- Topics and subscribers can be manually activated or deactivated using dedicated services.
- Subscribers that are no longer registered to the broker can be deleted.

Distributed Data Delivery (D3)

CHAPITRE 71

Introduction to D3

Ce chapitre contient les sections suivantes :

1. Overview
2. D3 terminology
3. Known limitations

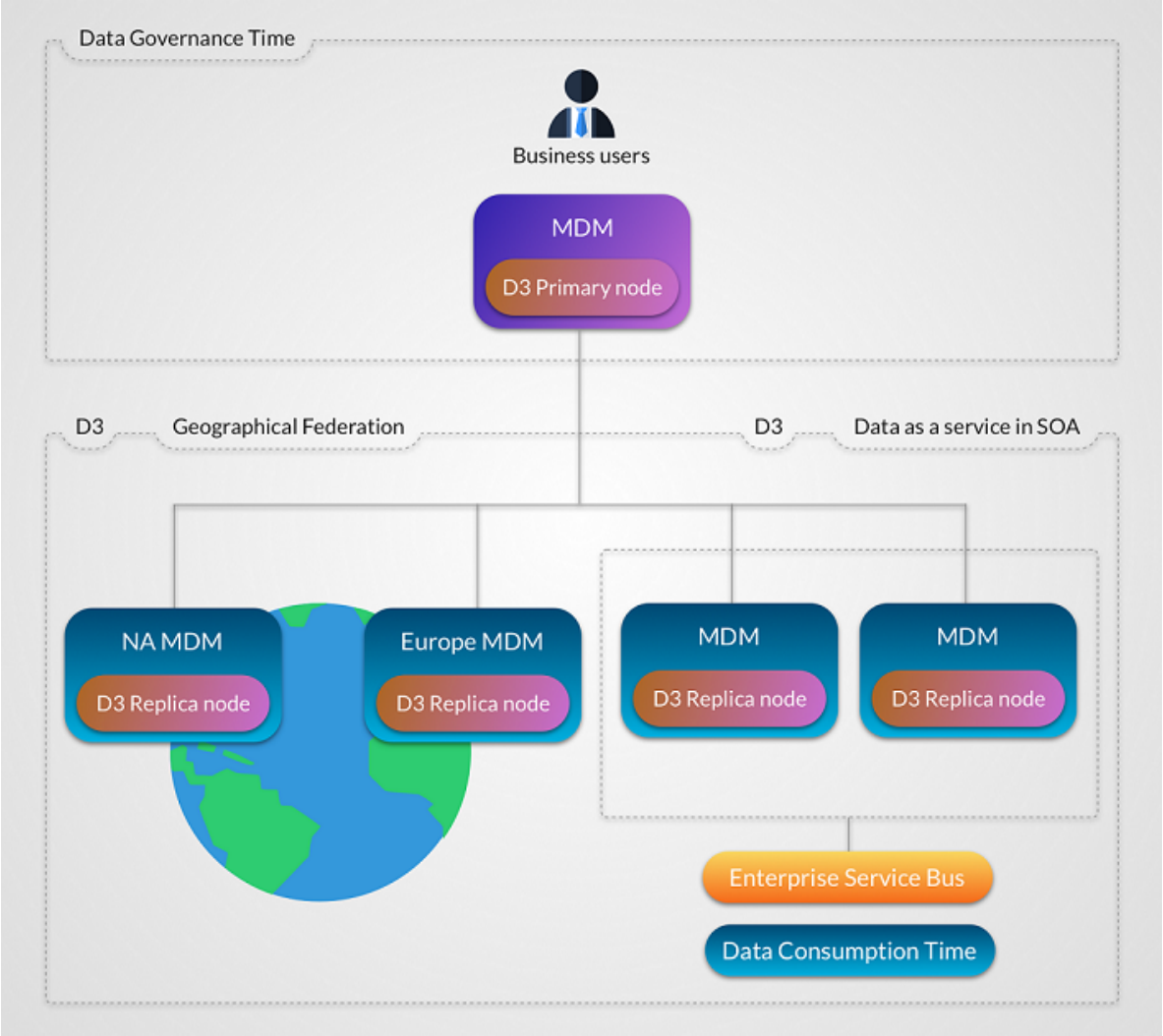
71.1 Overview

TIBCO EBX offers the ability to send data from an EBX instance to other instances. Using a broadcast action, it also provides an additional layer of security and control to the other features of EBX. It is particularly suitable for situations where data governance requires the highest levels of data consistency, approvals and the ability to rollback.

D3 architecture

A typical D3 installation consists of one primary node and multiple replica nodes. In the primary node, a Data Steward declares which dataspaces must be broadcast, as well as which user profile is allowed to broadcast them to the replica nodes. The Data Steward also defines delivery profiles, which are groups of one or more dataspaces.

Each replica node must define from which delivery profile it receives broadcasts.



Involving third-party systems

The features of D3 also allow third-party systems to access the data managed in EBX through data services. Essentially, when a system consumes the data of a delivery dataspace, the data is transparently redirected to the last broadcast snapshot. This ensures a more controlled and reliable view of the managed data.

Third-party systems can either access data directly through the primary node or through a replica node. Thus, a physical architecture consisting of a primary node and no replica nodes is possible.

Protocols

If JMS is activated, the conversation between a primary node and a replica node is based on SOAP over JMS, while archive transfer is based on JMS binary messages.

If JMS is not activated, conversation between a primary node and a replica node is based on SOAP over HTTP(S), while binary archive transfer is based on TCP sockets. If HTTPS is used, make sure that the target node connector is correctly configured by enabling SSL with a trusted certificate.

Voir aussi*JMS for distributed data delivery (D3)* *[p 457]*

71.2 D3 terminology

broadcast	Send a publication of an official snapshot of data from a primary node to replica nodes. The broadcast transparently and transactionally ensures that the data is transferred to the replica nodes.
delivery dataspace	A delivery dataspace is a dataspace that can be broadcast to authenticated and authorized users using a dedicated action. By default, when a data service accesses a delivery dataspace on any node, it is redirected to the last snapshot that was broadcast. See Data services [p 455].
delivery profile	A delivery profile is a logical name that groups one or more delivery dataspaces. Replica nodes subscribe to one or more delivery profiles.
cluster delivery mode	Synchronization with subscribed replica nodes is performed in a two-phase commit transactional process. This delivery mode is designed to respond to a high volume of queries using load balancing and/or fault tolerance. It ensures the consistency of data in the cluster between replica nodes and their primary node delivery dataspaces. Primary and replica nodes use the same last broadcast snapshots.
federation delivery mode	Synchronization is performed in a single phase, and with each registered replica node independently. This delivery mode is designed to be used with geographically distributed and/or heterogeneous architectures where response time and network availability cannot be guaranteed. At any one time, replica nodes can be at different last broadcast snapshots. The synchronization processes are thus independent of one another and replay of individual replica nodes are performed for certain broadcast failures.
Primary node	An instance of EBX that can define one or more delivery dataspaces, and to which replica nodes can subscribe. A primary node can also act as a regular EBX server.
Replica node	An instance of EBX attached to a primary node, in order to receive delivery dataspace broadcasts. Besides update restrictions on delivery dataspaces, the replica node acts as a regular EBX server.

Hub node	An instance of EBX acting as both a primary node and a replica node. Primary delivery dataspaces and replica node delivery dataspaces **must** be disjoint.

71.3 Known limitations

General limitations

- Each replica node must have only one primary node.
- Embedded data models cannot be used in D3 dataspaces. Therefore, it is not possible to create a dataset based on a publication in a D3 dataspace.
- The compatibility is not assured if at least one replica node product version is different from the primary node.

Broadcast and delivery dataspace limitations

- Access rights on dataspaces are not broadcast, whereas access rights on datasets are.
- Dataspace information is not broadcast.
- Dataspaces defined in relational mode cannot be broadcast.
- If a dataspace and its parent are broadcast, their parent-child relationship will be lost in the replica nodes.
- Once a snapshot has been broadcast to a replica, subsequent broadcasts of *any* snapshot with the same name will result in restoring the originally broadcast version of that same name on the replica node. That is, if the original snapshot on the primary node is purged and a new one is created with the same name and subsequently broadcast, then the content of the replica will be restored to that of the previously broadcast snapshot, and not to the latest one of the same name.
- To guarantee dataspace consistency between D3 nodes, the data model (embedded or packaged in a module) on which the broadcast contents are based, must be the same between the primary node and its replica nodes.
- On a replica delivery dataspace, if several replica nodes are registered, and if replication is enabled in data models, it will be effective for all nodes. No setting is available to activate/deactivate replication according to D3 nodes.
- Replication on replica nodes does not take part in the distributed transaction: it is automatically triggered after commit.

Administration limitations

Technical dataspaces cannot be broadcast, thus the EBX default user directory cannot be synchronized using D3.

CHAPITRE 72

D3 broadcasts and delivery dataspaces

Ce chapitre contient les sections suivantes :

1. Broadcast
2. Replica node registration
3. Accessing delivery dataspaces

72.1 Broadcast

Scope and contents of a broadcast

A D3 broadcast occurs at the dataspace or snapshot level. For dataspace broadcasts, D3 first creates a snapshot to capture the current state, then broadcasts this newly created snapshot.

A broadcast performs one of the following procedures depending on the situation:

- An update of the differences computed between the new broadcast snapshot and the current 'commit' one on the replica node.
- A full synchronization containing all datasets, tables, records, and permissions. This is done on the first broadcast to a given replica node, if the previous replica node commit is not known to the primary node, or on demand using the user service in '[D3] Primary node configuration'.

Voir aussi*Services on primary nodes [p 470]*

Performing a broadcast

The broadcast can be performed:

- By the end-user, using the **Broadcast** action available in the dataspace or snapshot (this action is available only if the dataspace is registered as a delivery dataspace)
- Using custom Java code that uses `D3NodeAsMaster`[API].

Conditions

In order to be able to broadcast, the following conditions must be fulfilled:

- The authenticated user profile has permission to broadcast.

- The dataspace or snapshot to be broadcast has no validation errors.

  Note: Although it is not recommended, it is possible to force a broadcast of a delivery dataspace that contains validation errors. In order to do this, set the maximum severity threshold allowed in a delivery dataspace validation report under '[D3] Primary node configuration' in the 'Administration' area.
- The D3 primary node configuration has no validation errors on the following scope: the technical record of the concerned delivery dataspace and all its dependencies (dependent delivery mappings, delivery profiles and registered replica nodes).
- The dataspace or snapshot does not contain any tables in relational mode.
- There is an associated delivery profile.
- If broadcasting a dataspace, the dataspace is not locked.
- If broadcasting a snapshot, the snapshot belongs to a dataspace declared as delivery dataspace and is not already the current broadcast snapshot (though a rollback to a previously broadcast snapshot is possible).
- The dataspace or snapshot contains differences compared to the last broadcast snapshot.

Persistence

When a primary node shuts down, all waiting or in progress broadcast requests abort, then they will be persisted on a temporary file. On startup, all aborted broadcasts are restarted.

Voir aussi*Temporary files* *[p 472]*

> **Note**
>
> Broadcasts are performed asynchronously. Therefore, no information is displayed in the user interface about the success or failure of a broadcast. Nevertheless, it is possible to monitor the broadcast operations inside '[D3] Primary node configuration'. See Supervision [p 471].

Destination

In the target replica or hub node side:

- The `ebx-d3-reference` dataspace identifier is the common parent of all the delivery dataspaces.
- The delivery dataspace has the same identifier in primary, replica or hub nodes.
- If the delivery dataspace is missing, it will be created on the first or on the full synchronization broadcast.
- If the delivery dataspace is already existing on the first broadcast or full synchronization it will be overridden.
- If an existing dataspace with the same identifier of the delivery one is detected outside of the `ebx-d3-reference`, An error will be raisen.

Voir aussi*Known limitations* *[p 452]*

72.2 Replica node registration

Scope and contents

An initialization occurs at the replica node level according to the delivery profiles registered in the TIBCO EBX main configuration file of the replica node. When the primary node receives that initialization request, it creates or updates the replica node entry, then sends the last broadcast snapshot of all registered delivery dataspaces.

> **Note**
>
> If the registered replica node repository ID or communication layer already exists, the replica node entry in the 'Registered replica nodes' technical table is updated, otherwise a new entry is created.

Performing an initialization

The initialization can be done:

- Automatically at replica node server startup.
- Manually when calling the replica node service 'Register replica node'.

Conditions

To be able to register, the following conditions must be fulfilled:

- The D3 mode must be 'hub' or 'slave'.
- The primary and replica node authentication parameters must correspond to the primary node administrator and replica node administrator defined in their respective directories.
- The delivery profiles defined on the replica node must exist in the primary node configuration.
- All data models contained in the registered dataspaces must exist in the replica node. If embedded, the data model names must be the same. If packaged, they must be located at the same module name and the schema path in the module must be the same in both the primary and replica nodes.
- The D3 primary node configuration has no validation error on the following scope: the technical record of the registered replica node and all its dependencies (dependent delivery profiles, delivery mappings and delivery dataspaces).

> **Note**
>
> To set the parameters, see the replica or hub EBX properties in Configuring primary, hub and replica nodes [p 467].

72.3 Accessing delivery dataspaces

Data services

By default, when a data service accesses a delivery dataspace, it is redirected to the current snapshot, which is the last broadcast one. However, this default behavior can be modified either at the request level or in the global configuration.

Voir aussi*Common parameter 'disableRedirectionToLastBroadcast'* *[p 642]*

Access restrictions

On the primary node, a delivery dataspace can neither be merged nor closed. Other operations are available depending on permissions. For example, modifying a delivery dataspace directly, creating a snapshot independent from a broadcast, or creating and merging a child dataspace.

On the replica node, aside from the broadcast process, no modifications of any kind can be made to a delivery dataspace, whether by the end-user, data services, or a Java program. Furthermore, any dataspace-related operations, such as merge, close, etc., are forbidden on the replica node.

D3 broadcast Java API

The last broadcast snapshot may change between two calls if a broadcast has taken place in the meantime. If a fully stable view is required for several successive calls, these calls need to specifically refer to the same snapshot.

To get the last broadcast snapshot, see `D3Node.getBroadcastVersion`[API].

CHAPITRE 73

D3 JMS Configuration

Ce chapitre contient les sections suivantes :

1. JMS for distributed data delivery (D3)

73.1 JMS for distributed data delivery (D3)

To configure D3 to use JMS instead of the default HTTP and TCP protocols, you must configure the JMS connection factory [p 348] and the following queues declared in the `WEB-INF/web.xml` deployment descriptor of the '`ebx`' web application.

> **Note**
>
> If the TIBCO EBX main configuration does not activate JMS and D3 ('slave', 'hub' or 'master' node) through the properties `ebx.d3.mode`, `ebx.jms.activate` and `ebx.jms.d3.activate`, then the environment entries below will be ignored by EBX runtime. See JMS [p 379] and Distributed data delivery (D3) [p 379] in the EBX main configuration properties for more information on these properties.

Common declarations on primary and replica nodes (for shared queues)

Reserved resource name	Default JNDI name	Description
`jms/EBX_D3MasterQueue`	Weblogic: `EBX_D3MasterQueue` JBoss: `java:/jms/EBX_D3MasterQueue`	D3 primary JMS queue (only for D3 mode 'slave' or 'hub'). It specifies the queue name used to send SOAP requests to the D3 primary node. The message producer sets the primary node repository ID as a value of the header field `JMSType`. Java type: `javax.jms.Queue`
`jms/EBX_D3ReplyQueue`	Weblogic: `EBX_D3ReplyQueue` JBoss: `java:/jms/EBX_D3ReplyQueue`	D3 Reply JMS queue (for all D3 modes except the 'single' mode). It specifies the name of the reply queue for receiving SOAP responses. The consumption is filtered using the header field `JMSCorrelationID`. Java type: `javax.jms.Queue`
`jms/EBX_D3ArchiveQueue`	Weblogic: `EBX_D3ArchiveQueue` JBoss: `java:/jms/EBX_D3ArchiveQueue`	D3 JMS Archive queue (for all D3 modes except the 'single' mode). It specifies the name of the transfer archive queue used by the D3 node. The consumption is filtered using the header field `JMSCorrelationID`. If the archive weight is higher than the threshold specified in the property `ebx.jms.d3.archiveMaxSizeInKB`, the archive will be divided into several sequences. Therefore, the consumption is filtered using the header fields `JMSXGroupID` and `JMSXGroupSeq` instead. Java type: `javax.jms.Queue`
`jms/EBX_D3CommunicationQueue`	WebLogic: `EBX_D3CommunicationQueue` JBoss: `java:/jms/EBX_D3CommunicationQueue`	D3 JMS Communication queue (for all D3 modes except 'single' mode). It specifies the name of the communication queue where the requests are received. The consumption is filtered using the header field `JMSType` which corresponds to the current repository ID. Java type: `javax.jms.Queue`

> **Note**
>
> These JNDI names are set by default, but can be modified inside the web application archive `ebx.war`, included in `EBXForWebLogic.ear` (if using Weblogic) or in `EBX.ear` (if using JBoss, Websphere or other application servers).

Optional declarations on primary nodes (for replica-specific queues)

> **Note**
>
> Used for ascending compatibility prior to 5.5.0 or for mono-directional queues topology.

The deployment descriptor of the primary node must be manually modified by declaring specific communication and archive queues for each replica node. It consists in adding resource names in 'web.xml' inside 'ebx.war'. The replica-specific node queues can be used by one or more replica nodes.

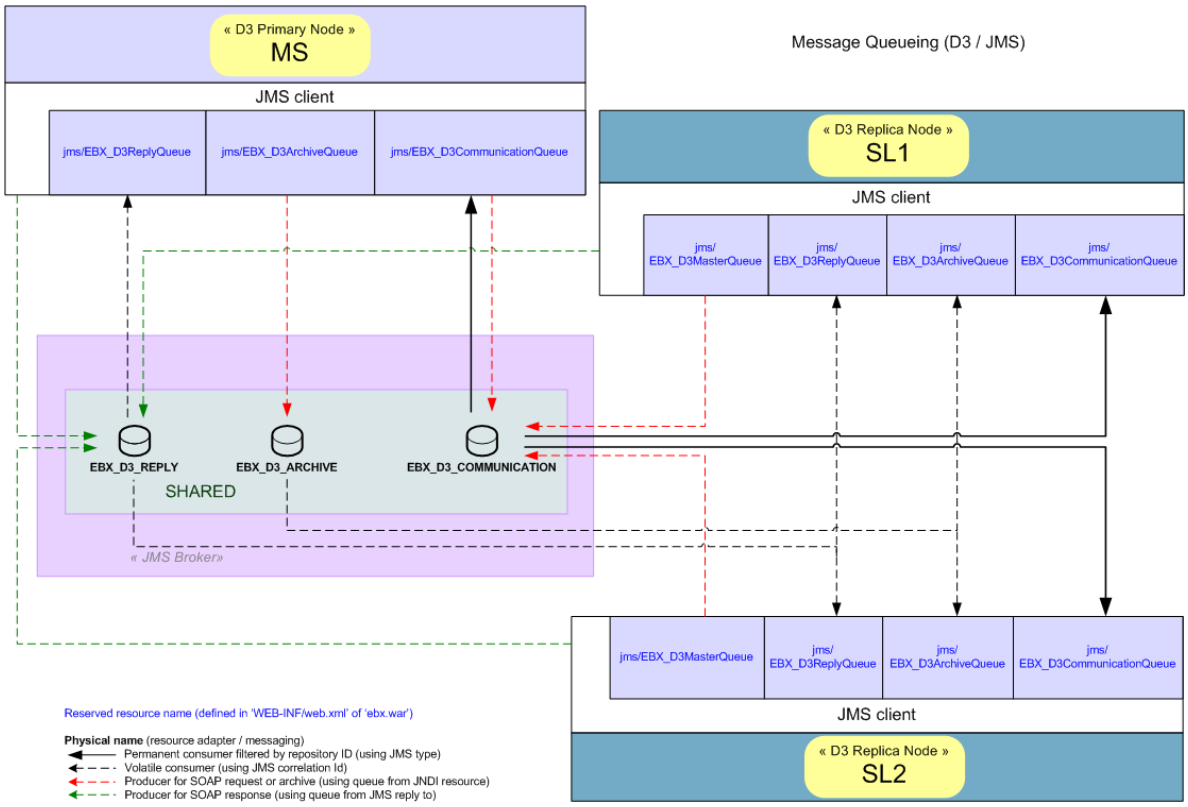
Resources can be freely named, but the physical names of their associated queue must correspond to the definition of replica nodes for resources `jms/EBX_D3ArchiveQueue` and `jms/EBX_D3CommunicationQueue`.

> **Note**
>
> Physical queue names matching: on registration, the replica node sends the communication and archive physical queue names. These queues are matched by physical queue name among all resources declared on the primary node. If unmatched, the registration fails.

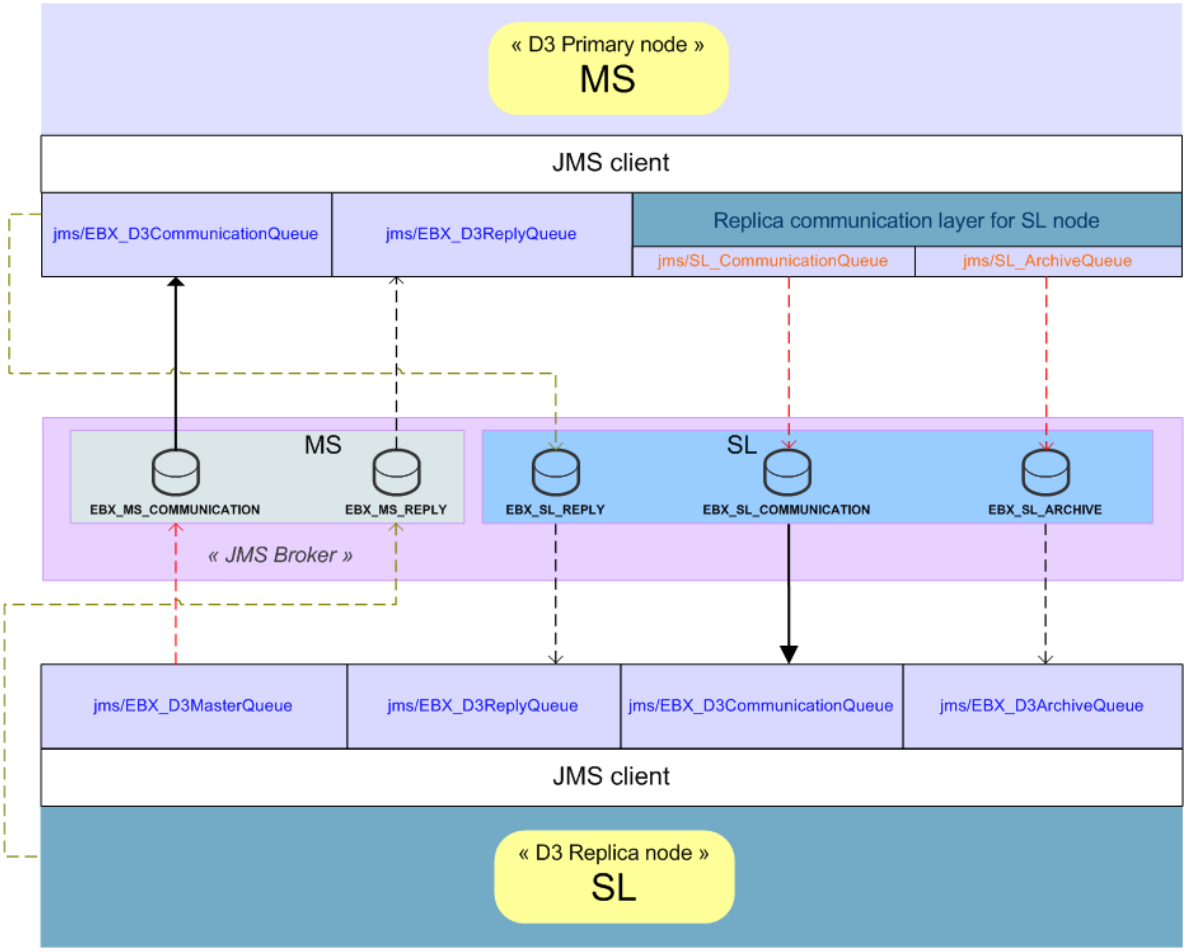
Examples of JMS configuration

	Shared queues	**Specific queues**
Primary-Replica nodes architecture	Between a primary node and two replica nodes with shared queues [p 460]	Between a primary node and a replica node with replica-specific queues [p 461]
Hub-Hub architecture	Between two hub nodes with shared queues [p 462]	Between two hub nodes with replica-specific queues [p 463]

Between a primary node and two replica nodes with shared queues



Between a primary node and a replica node with replica-specific queues

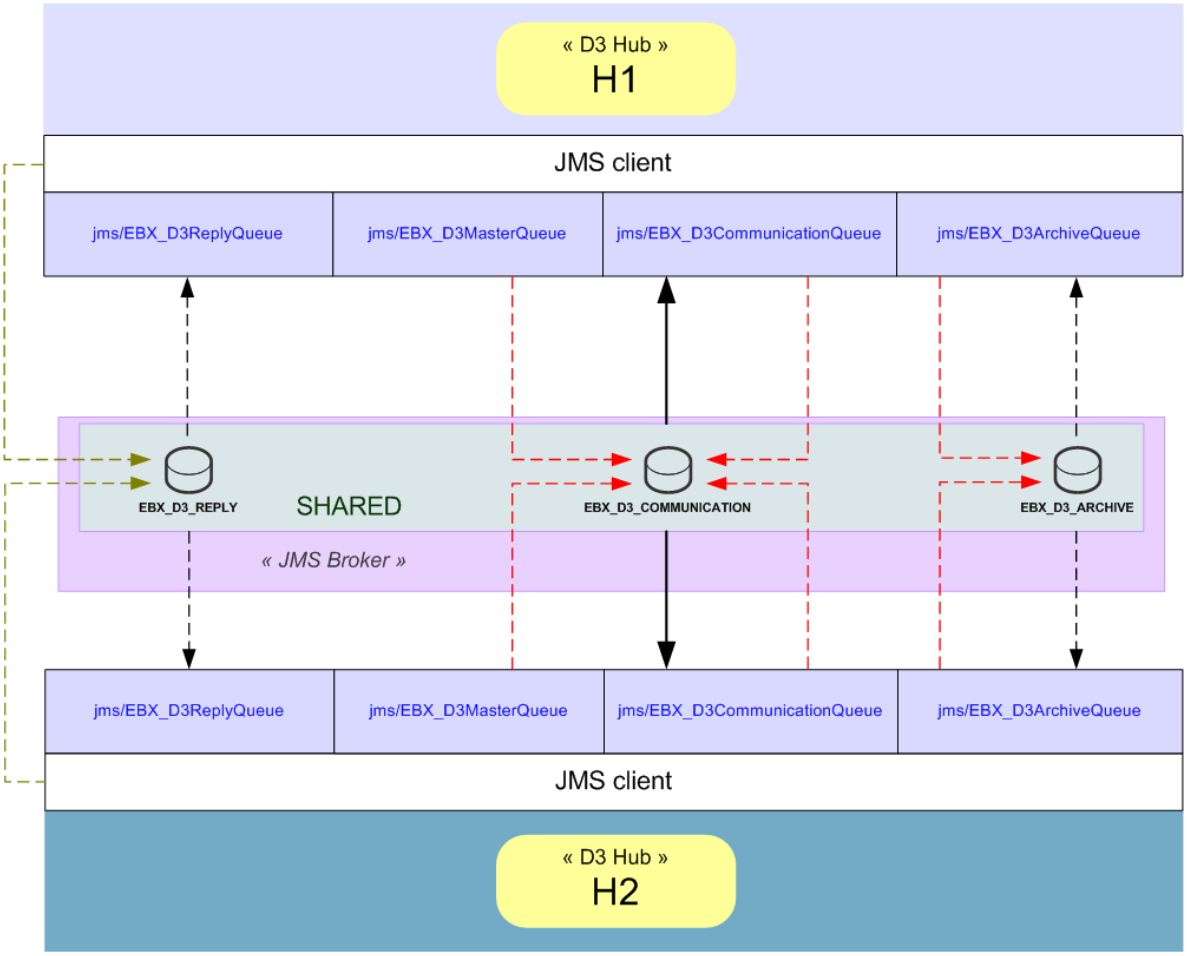


Reserved or custom resource name (defined in 'WEB-INF/web.xml' of 'ebx.war')

Physical name (resource adapter / messaging)

- Permanent consumer filtered by repository ID (using JMS type)
- Volatile consumer (using JMS correlation Id)
- Producer for SOAP request or archive (using queue from JNDI resource)
- Producer for SOAP response (using queue from JMS reply to)

Between two hub nodes with shared queues

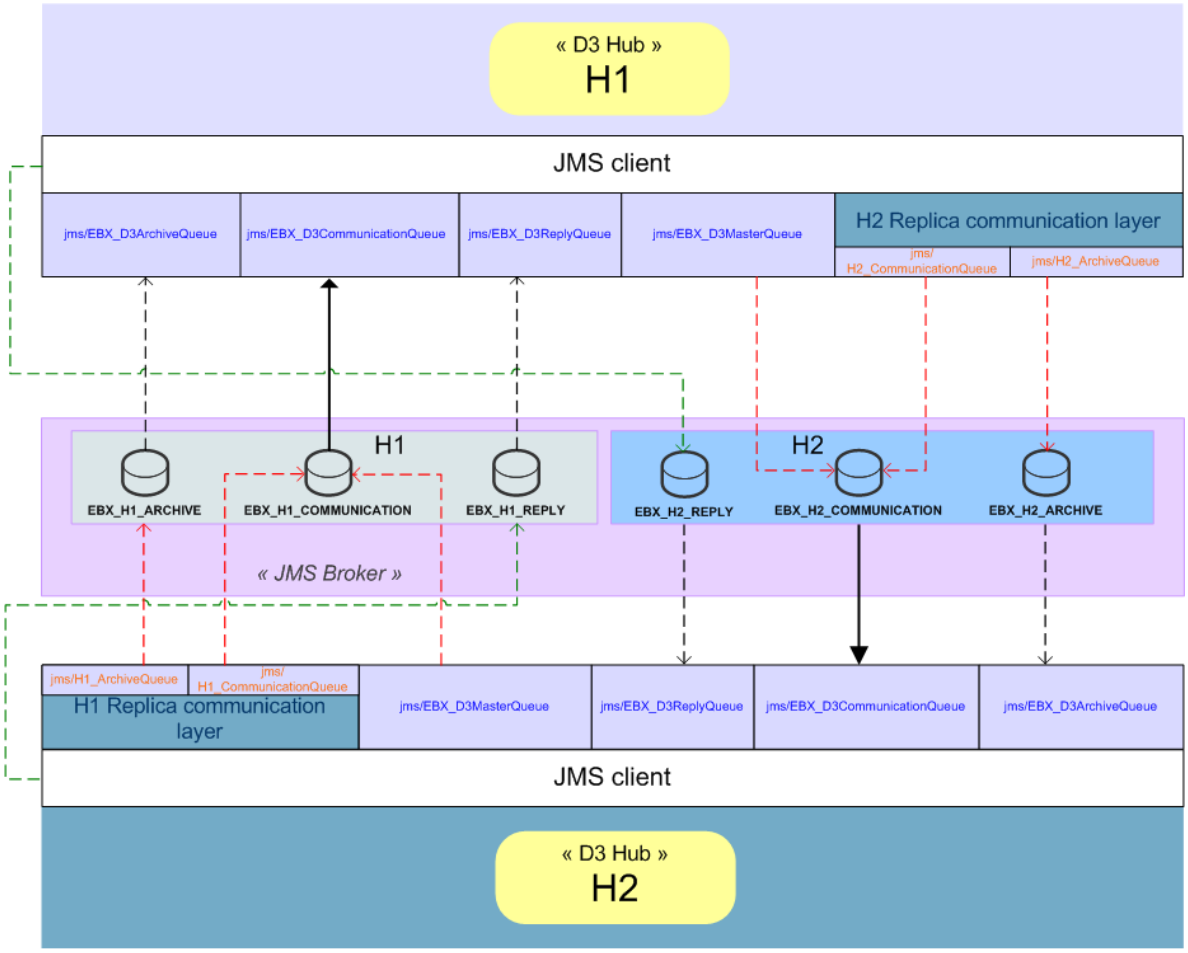


Reserved resource name (defined in 'WEB-INF/web.xml' of 'ebx.war')

Physical name (resource adapter / messaging)

- Permanent consumer filtered by repository ID (using JMS type)
- Volatile consumer (using JMS correlation Id)
- Producer for SOAP request or archive (using queue from JNDI resource)
- Producer for SOAP response (using queue from JMS reply to)

Between two hub nodes with replica-specific queues



Reserved or custom resource name (defined in 'WEB-INF/web.xml' of 'ebx.war')

Physical name (resource adapter / messaging)

- Permanent consumer filtered by repository ID (using JMS type)
- Volatile consumer (using JMS correlation Id)
- Producer for SOAP request or archive (using queue from JNDI resource)
- Producer for SOAP response (using queue from JMS reply to)

CHAPITRE 74

D3 administration

Ce chapitre contient les sections suivantes :

1. Quick start
2. Configuring D3 nodes
3. Supervision

74.1 Quick start

This section introduces the configuration of a basic D3 architecture with two TIBCO EBX instances. Before starting, please check that each instance can work properly with its own repository.

> **Note**
>
> Deploy EBX on two different web application containers. If both instances are running on the same host, ensure that all communication TCP ports are distinct.

Declare an existing dataspace on the primary node

The objective is to configure and broadcast an existing dataspace from a *primary* node.

This configuration is performed on the entire D3 infrastructure (primary [p 451] and replica [p 451] nodes included).

Update the `ebx.properties`*primary* node configuration file with:

1. Define D3 mode as `primary` in key `ebx.d3.mode`.

> **Note**
>
> The *primary* node can be started after the configuration.

After authenticating as a built-in administrator, navigate within the administration tab:

1. Prerequisite: Check that the node is configured as a *primary* node (in the 'Actions' menu use 'System information' and check 'D3 mode').
2. Open the '[D3] Primary configuration' administration feature.
3. Add the dataspace to be broadcast to the 'Delivery dataspaces' table, and declare the allowed profile.
4. Add the delivery profile [p 451] to the 'Delivery profiles' table (it must correspond to a logical name) and declare the delivery mode. Possible values are: cluster mode [p 451] or federation mode [p 451].

5. Map the delivery dataspace with the delivery profile into the 'Delivery mapping' table.

> **Note**
>
> The *primary* node is now ready for the replica node(s) registration on the delivery profile.
>
> Check that the D3 broadcast menu appears in the 'Actions' menu of the dataspace or one of its snapshots.

Configure replica node for registration

The objective is to configure and register the *replica* node based on a delivery profile and communications settings.

Update the `ebx.properties` replica node configuration file with:

1. Define D3 mode as `replica` in key `ebx.d3.mode`.
2. Define the delivery profile [p 451] set on the *primary* node in key `ebx.d3.delivery.profiles` (delivery profiles must be separated by a comma and a space).
3. Define the *primary* node user authentication (must have the built-in administrator profile) for node communications in `ebx.d3.master.username` and `ebx.d3.master.password`.
4. Define HTTP/TCP protocols [p 468] for *primary* node communication, by setting a value for the property key `ebx.d3.master.url`

   (for example `http://localhost:8080/ebx-dataservices/connector`).
5. Define the *replica* node user authentication (must have the built-in administrator profile) for node communications in `ebx.d3.slave.username` and `ebx.d3.slave.password`.
6. Define HTTP/TCP protocols [p 468] for *replica* node communication, by setting a value for the property key `ebx.d3.slave.url`

   (for example `http://localhost:8090/ebx-dataservices/connector`).

> **Note**
>
> The *replica* node can be started after the configuration.

After authenticating as a built-in administrator, navigate inside the administration tab:

1. Prerequisite: Check that the node is configured as the *replica* node (in the 'Actions' menu use 'System information' and check 'D3 mode').
2. Open the '[D3] Replica configuration' administration feature.
3. Check the information on the 'Primary information' screen: No field should have the 'N/A' value.

> **Note**
>
> Please check that the model is available before broadcast (from data model assistant, it must be published).
>
> The *replica* node is then ready for broadcast.

74.2 Configuring D3 nodes

Runtime configuration of primary and hub nodes through the user interface

The declaration of delivery dataspaces and delivery profiles is done by selecting the '[D3] Primary configuration' feature from the 'Administration' area, where you will find the following tables:

Delivery dataspaces	Declarations of the dataspaces that can be broadcast.
Delivery profiles	Profiles to which replica nodes can subscribe. The delivery mode must be defined for each delivery profile.
Delivery mapping	The association between delivery dataspaces and delivery profiles.

> **Note**
>
> The tables above are read-only while some broadcasts are pending or in progress.

Configuring primary, hub and replica nodes

This section details how to configure a node in its EBX main configuration file.

Voir aussi*Overview* *[p 369]*

Primary node

In order to act as a *primary* node, an instance of EBX must declare the following property in its main configuration file.

Sample configuration for `ebx.d3.mode=master` node:

```
######################################################################
## D3 configuration
######################################################################
######################################################################
# Configuration for master, hub and slave
######################################################################
# Optional property.
# Possibles values are single, master, hub, slave
# Default is single meaning the server will be a standalone instance.
ebx.d3.mode=master
```

Voir aussi*primary node* *[p 451]*

Hub node

In order to act as a *hub* node (combination of primary and replica node configurations), an instance of EBX must declare the following property in its main configuration file.

Sample configuration for `ebx.d3.mode=hub` node:

```
######################################################################
## D3 configuration
######################################################################
######################################################################
# Configuration for master, hub and slave
```

```
######################################################################
# Optional property.
# Possibles values are single, master, hub, slave
# Default is single meaning the server will be a standalone instance.
ebx.d3.mode=hub

######################################################################
# Configuration dedicated to hub or slave
######################################################################
# Profiles to subscribe to
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.delivery.profiles=

# User and password to be used to communicate with the master.
# Mandatory properties if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.master.username=
ebx.d3.master.password=

# User and password to be used by the master to communicate with the hub or slave.
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.slave.username=
ebx.d3.slave.password=
```

Voir aussi*hub node* *[p 452]*

Replica node

In order to act as a *replica* node, an instance of EBX must declare the following property in its main configuration file.

Sample configuration for `ebx.d3.mode=slave` node:

```
######################################################################
## D3 configuration
######################################################################
######################################################################
# Configuration for master, hub and slave
######################################################################
# Optional property.
# Possibles values are single, master, hub, slave
# Default is single meaning the server will be a standalone instance.
ebx.d3.mode=slave

######################################################################
# Configuration dedicated to hub or slave
######################################################################
# Profiles to subscribe to
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.delivery.profiles=

# User and password to be used to communicate with the master.
# Mandatory properties if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.master.username=
ebx.d3.master.password=

# User and password to be used by the master to communicate with the hub or slave.
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave
ebx.d3.slave.username=
ebx.d3.slave.password=
```

Voir aussi*replica node* *[p 451]*

Configuring the network protocol of a node

This section details how to configure the network protocol of a node in its EBX main configuration file.

Voir aussi*Overview* *[p 369]*

HTTP(S) and socket TCP protocols

Sample configuration for `ebx.d3.mode=hub` or `ebx.d3.mode=slave` node with HTTP(S) network protocol:

```
######################################################################
```

```
# HTTP(S) and TCP socket configuration for D3 hub and slave
#####################################################################
# URL to access the data services connector of the master
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave and JMS for D3 is not activated.
# This property will be ignored if JMS for D3 is activated.
# The URL must follow this pattern: [protocol]://[master_host]:[master_port]/ebx-dataservices/connector
# Where the possible values of 'protocol' are 'http' or 'https'.
ebx.d3.master.url=

# URL to access the data services connector of the slave
# Mandatory property if ebx.d3.mode=hub or ebx.d3.mode=slave and JMS for D3 is not activated.
# This property will be ignored if JMS for D3 is activated.
# The URL must follow this pattern: [protocol]://[slave_host]:[slave_port]/ebx-dataservices/connector
# Where the possible values of 'protocol' are 'http' or 'https'.
ebx.d3.slave.url=

# Minimum port to use to transfer archives on TCP mode.
# Must be a positive integer above zero and below 65535.
# If not set, a random port will be used.
#ebx.d3.slave.socket.range.min=

# Max port to use on TCP mode to transfer archives.
# Must be a positive integer above ebx.d3.slave.socket.range.min and below 65535.
# Mandatory if ebx.d3.slave.socket.range.min is set.
#ebx.d3.slave.socket.range.max=
```

JMS protocol

If JMS is activated, the following properties can be defined in order to enable JMS functionalities for a D3 node.

Sample configuration for all D3 nodes with JMS network protocol:

```
#####################################################################
## JMS configuration for D3
#####################################################################
# Taken into account only if Data Services JMS is configured properly
#####################################################################
# Configuration for master, hub and slave
#####################################################################
# Default is false, activate JMS for D3
## If activated, the deployer must ensure that the entries
## 'jms/EBX_D3ReplyQueue', 'jms/EBX_D3ArchiveQueue' and 'jms/EBX_D3CommunicationQueue'
## are bound in the operational environment of the application server.
## On slave or hub mode, the entry 'jms/EBX_D3MasterQueue' must also be bound.
ebx.jms.d3.activate=false

# Change the default timeout when using reply queue.
# Must be a positive integer that does not exceed 3600000.
# Default is 10000 milliseconds.
#ebx.jms.d3.reply.timeout=10000

# Time-to-live message value expressed in milliseconds.
# This value will be set on each message header 'JMSExpiration' that defines the
# countdown before the message deletion managed by the JMS broker.
# Must be a positive integer equal to 0 or above the value of 'ebx.jms.d3.reply.timeout'.
# The value 0 means that the message does not expire.
# Default is 3600000 (one hour).
#ebx.jms.d3.expiration=3600000

# Archive maximum size in KB for the JMS body message. If exceeds, the message
# is transferred into several sequences messages in a same group, where each one does
# not exceed the maximum size defined.
# Must be a positive integer equals to 0 or above 100.
# Default is 0 that corresponds to unbounded.
#ebx.jms.d3.archiveMaxSizeInKB=

#####################################################################
# Configuration dedicated to hub or slave
#####################################################################
# Master repository ID, used to set a message filter for the concerned master when sending JMS message
# Mandatory property if ebx.jms.d3.activate=true and if ebx.d3.mode=hub or ebx.d3.mode=slave
#ebx.jms.d3.master.repositoryId=
```

Voir aussi*JMS for distributed data delivery (D3)* *[p 457]*

Services on primary nodes

Services to manage a primary node are available in the 'Administration' area of the replica node under '[D3] Primary node configuration' and also in the 'Delivery dataspaces' and 'Registered replica nodes' tables. The services are:

Relaunch replays	Immediately relaunch all replays for waiting federation deliveries.
Delete replica node delivery dataspace	Delete the delivery dataspace on chosen replica nodes and/or unregister it from the configuration of the D3 primary node. To access the service, select a delivery dataspace from the 'Delivery dataspaces' table on the primary node, then launch the wizard.
Fully resynchronize	Broadcast the full content of the last broadcast snapshot to the registered replica nodes.
Subscribe a replica node	Subscribe a set of selected replica nodes.
Deactivate replica nodes	Remove the selected replica nodes from the broadcast scope and switch their states to 'Unavailable'. **Note** The "in progress" broadcast contexts are rolled back.
Unregister replica nodes	Disconnects the selected replica nodes from the primary node. **Note** The "in progress" broadcast contexts are rolled back.

Note

The primary node services above are hidden while some broadcasts are pending or in progress.

Services on replica nodes

Services are available in the 'Administration' area under *[D3] Configuration of replica* node to manage its subscription to the primary node and perform other actions:

Register replica node	Re-subscribes the replica node to the primary node if it has been unregistered.
Unregister replica node	Disconnects the replica node from the primary node. **Note** The "in progress" broadcast contexts are rolled back.
Close and delete snapshots	Clean up a replica node delivery dataspace. To access the service, select a delivery dataspace from the 'Delivery dataspaces' table on the replica node, then follow the wizard to close and delete snapshots based on their creation dates. **Note:** The last broadcast snapshot is automatically excluded from the selection.

74.3 Supervision

The last broadcast snapshot is highlighted in the snapshot table of the dataspace, it is represented by an icon displayed in the first column.

Primary node management console

Several tables make up the management console of the primary node, located in the 'Administration' area of the primary node, under '[D3] Primary node configuration'. They are as follows:

Registered replica nodes	Replica nodes registered with the primary node. From this table, several services are available on each record.
Broadcast history	History of broadcast operations that have taken place.
Replica node registration log	History of initialization operations that have taken place.
Detailed history	History of archive deliveries that have taken place. The list of associated delivery archives can be accessed from the tables 'Broadcast history' and 'Initialization history' using selection nodes.

Primary node supervision services

Available in the 'Administration' area of the primary node under '[D3] Primary node configuration'. The services are as follows:

Check replica node information	Lists the replica nodes and related information, such as the replica node's state, associated delivery profiles, and delivered snapshots.
Clear history content	Deletes all records in all history tables, such as 'Broadcast history', 'Replica node registration log' and 'Detailed history'.

Replica node monitoring through the Java API

A replica node monitoring class can be created to implement actions that are triggered when the replica node's status switches to either 'Available' or 'Unavailable'. To do so, it must implement the `NodeMonitoring` interface. This class must be outside of any EBX module and accessible from the class-loader of 'ebx.jar' and its full class name must be specified under '[D3] Replica node configuration'.

Voir aussi *`NodeMonitoring`*[API]

Primary node notification

A D3 administrator can set up mail notifications to receive broadcast events:

- On broadcast failure,
- On federation broadcast, if replays exceed a given threshold.

The mail contains a table of events with optional links to further details.

To enable notifications, open the '[D3] Primary node configuration' dataspace from the 'Administration' area and configure the 'Notifications' group under 'Global configuration'.

The 'From email' and 'URL definition' options should also be configured by using the 'Email configuration' link.

Log supervision

The technical supervision can be done through the log category 'ebx.d3', declared in the EBX main configuration file. For example:

```
ebx.log4j.category.log.d3= INFO, Console, ebxFile:d3
```

Voir aussi *Configuring the EBX logs* *[p 375]*

Temporary files

Some temporary files, such as exchanged archives, SOAP messages, broadcast queue, (...), are created and written to the EBX temporary directory. This location is defined in the EBX main configuration file:

```
################################################
```

```
## Directories for temporary resources.
################################################
# When set, allows specifying a directory for temporary files different from java.io.tmpdir.
# Default value is java.io.tmpdir
ebx.temp.directory = \\${java.io.tmpdir}

# Allows specifying the directory containing temporary files for cache.
# If unset, the used directory is ${ebx.temp.directory}/ebx.platform.
#ebx.temp.cache.directory = ${ebx.temp.directory}/ebx.platform

# When set, allows specifying the directory containing temporary files for import.
# If unset, the used directory is ${ebx.temp.directory}/ebx.platform.
#ebx.temp.import.directory = ${ebx.temp.directory}/ebx.platform
```

Guide de sécurité (en anglais)

CHAPITRE 75

Security Best Practices

Here is a list of best practices that are considered useful to enforce a good security level for the EBX setup. These best practices apply to EBX and to other environments, their configuration, protocols and policies. These are best practices in general, and may not be relevant to your particular infrastructure and security policy.

Ce chapitre contient les sections suivantes :

1. Encryption algorithms
2. HTTPS
3. Installation
4. Web Server
5. Application Server
6. Java
7. Database
8. User directory and Administration rights

75.1 Encryption algorithms

Web Server or Application Server may specify encryption algorithms when setting HTTPS parameters. Some recommendations on these algorithms are provided in section HTTPS [p 476]. Password and fields having osd:password as a type are storing hash of their value with SHA_512 as algorithm. That includes the password of users of the default directory.

75.2 HTTPS

Using HTTPS for communication with clients (GUI and REST or SOAP) is recommended. All HTTP traffic should be redirected to HTTPS.

A secure cipher suite and protocols should be used whenever possible. This applies, for example, to Web Servers, Application Servers, and jdbc connections.

TLS v1.2 should be the main protocol because it's the only version that offers modern authenticated encryption (also known as AEAD).

Several obsolete cryptographic primitives must be avoided:

- Anonymous Diffie-Hellman (ADH) suites do not provide authentication,

- NULL cipher suites provide no encryption,
- Export cipher suites are insecure when negotiated in a connection, but they can also be used against a server that prefers stronger suites (the FREAK attack),
- Suites with weak ciphers (typically of 40 and 56 bits) use encryption that can easily be broken,
- RC4 is insecure,
- 3DES is slow and weak,

On the other hand, getting too restrictive on allowed cyphers may prevent some clients to connect as they may not be able to negotiate the HTTPS connection.

The following configuration is compatible with browsers supported by EBX.

- Cipher suites: ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256
- Versions: TLSv1.2

75.3 Installation

Deployed components as Web Server and Application Server should be installed using a non-root or unprivileged user, and following the principle of least privilege whenever possible. For example, only necessary ports and protocols should be opened.

75.4 Web Server

If you have to expose web applications on the Internet, it's a good practice to protect them with a Web Server in a demilitarized zone while EBX and the database server may be in a production zone. Here are some best practices for the configuration.

The secure cipher suite and protocols should be set according to the above section HTTPS [p 476].

It is also a best practice not to use the default configuration, and to remove any banner that may also expose the version and type of web server.

For example, on Apache2, to remove the banner (default page returned at the root), just remove the folder `/var/www/html`.

Also, on Apache2, to remove headers identifying the Web Server, the value of `ServerTokens` and `ServerSignature` from the file `security.conf` should have the following values:

```
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
ServerTokens Prod

# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
ServerSignature Off
```

The Web Server is the recommended way for setting restrictions with HTTP security headers. Be aware that headers related to the origin will impact authorized URLs for all resources returned by EBX. That includes the content of fields of the URL type (example: image of avatar).

Here is a list of security headers and how to set them for EBX. First, EBX should be configured to not set any HTTP security headers. To do so, the property `ebx.security.headers.activated` must be set to 'false'.

X-XSS-Protection

The `x-xss-protection` header is designed to enable the cross-site scripting (XSS) filter built into modern web browsers. Here is what the header should look like.

```
x-xss-protection: 1; mode=block
```

For version 5.9.4, if the property `ebx.security.headers.activated` is not set or set to `true`, the security header must also be unset beforehand. For version 5.9.4, if the property `ebx.security.headers.activated` is set to `false`, the security header does not need to be unset, so ignore the first line in following snippets. For previous versions, the security header must be unset beforehand.

Enable in Nginx

```
header always unset x-xss-protection
header always set x-xss-protection "1; mode=block"
```

Enable in Apache2

```
proxy_hide_header x-xss-protection;
add_header x-xss-protection "1; mode=block" always;
```

x-Frame-Options

The `x-frame-options` header provides clickjacking protection by not allowing iframes to load on the site. Be aware, this may not be compatible with your configuration if EBX is integrated through frames for example. Here is what the header should look like:

```
x-frame-options: SAMEORIGIN
```

Enable in Nginx

```
add_header x-frame-options "SAMEORIGIN" always;
```

Enable in Apache2

```
header always sets x-frame-options "SAMEORIGIN"
```

X-Content-Type-Options

The `x-content-type-options` header prevents Internet Explorer and Google Chrome from sniffing a response away from the declared content-type. This helps reduce the danger of drive-by downloads and helps treat the content properly. Here is what the header looks like.

```
x-content-type-options: nosniff
```

Enable in Nginx

```
add_header X-Content-Type-Options "nosniff" always;
```

Enable in Apache2

```
header always sets X-Content-Type-Options "nosniff"
```

Strict-Transport-Security

The `strict-transport-security` header is a security enhancement that restricts web browsers to access web servers solely over HTTPS. This ensures the connection cannot be established through

an insecure HTTP connection which could be vulnerable to attacks. Here is what the header should look like:

```
strict-transport-security: max-age=31536000; includeSubDomains
```

Enable in Nginx

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

Enable in Apache2

```
header always sets Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

Content-Security-Policy

The `content-security-policy` HTTP header provides an additional layer of security. This policy helps prevent attacks such as Cross Site Scripting (XSS) and other code injection attacks by defining content sources which are approved and thus allowing the browser to load them. Here is what the header shuould look like. Make sure to adapt it with your domain name (`server.company.com` in the example).

```
content-security-policy: default-src 'self'; font-src * data: server.company.com; img-
src * data: server.company.com; script-src * 'unsafe-inline' 'unsafe-eval'; style-src
 * 'unsafe-inline';
```

Enable in Nginx

```
add_header Content-Security-Policy "default-src 'self'; font-src * data:
server.company.com; img-src * data: server.company.com; script-src * 'unsafe-inline'
'unsafe-eval'; style-src * 'unsafe-inline';" always;
```

Enable in Apache2

```
header always sets Content-Security-Policy "default-src 'self'; font-src * data:
server.company.com; img-src * data: server.company.com; script-src * 'unsafe-inline'
'unsafe-eval'; style-src * 'unsafe-inline';"
```

Referrer-Policy

The Referrer-Policy HTTP header governs which referrer information should be included with requests made. The Referrer-Policy tells the web browser how to handle referrer information that is sent when a user clicks on a link that leads to another page. Here is what it should look like:

```
Referrer-Policy: strict-origin
```

Enable in Nginx

```
add_header Referrer-Policy: "strict-origin" always;
```

Enable in Apache2

```
header always sets Referrer-Policy "strict-origin"
```

75.5 Application Server

As for Web Servers, the same best practice applies: do not expose technical information on the Application Server. For example, for Tomcat, it is recommended to fill the attribute `server` of `connector` in `server.xml` with a generic value as `AppServer`.

```
<Connector port="8080" enableLookups="false" protocol="HTTP/1.1" useBodyEncodingForURI="true"
server="AppServer"/>
```

If the Application Server is exposed through HTTPS, the secure cipher suite and Protocols should be set according to the above section HTTPS [p 476].

If there is a Web Server, it is also recommended to use ports higher than 1024 and let the Web Server do proxy.

If there is no Web Server, security headers should be set by the Application Server as described above.

75.6 Java

It is recommended to follow the security best practices from Oracle. Last supported patches should also be applied as soon as they are available especially when they include security patches. Consider using the Server JRE for server systems, such as application servers or other long-running back-end processes. The Server JRE is the same as the regular JRE except that it does not contain the web-browser plugins.

75.7 Database

Databases should be encrypted at rest and in transit. If there is a private key for encryption, it should not be stored in the same location as the data files. Regarding the JDBC connection, consider configuring the JDBC driver to use SSL/TLS. Contact your database administrator for detailed instructions. You should always use the last supported version or RDBMS including drivers.

75.8 User directory and Administration rights

For production and test platforms, EBX must be integrated with a custom directory [p 426] to enforce the password policy of your company. The default directory can be used only for development platforms.

According to the Separation of Duties best practice, administrators can manage users and grant access but should not have any functional rights.

Guide du développeur (en anglais)

Introduction

CHAPITRE 76

Packaging TIBCO EBX modules

An EBX module is a standard Java EE web application, packaging various resources such as XML Schema documents, Java classes and static resources.

Since EBX modules are web applications they benefit from features such as class-loading isolation, WAR or EAR packaging, and Web resources exposure.

Ce chapitre contient les sections suivantes :

1. Module structure
2. Module declaration
3. Module registration
4. Packaged resources

76.1 Module structure

An EBX module contains the following files:

`/WEB-INF/ebx/module.xml`	This mandatory document defines the main properties and services of the module. See Module declaration [p 484].
`/WEB-INF/web.xml`	This is the standard Java EE deployment descriptor. It can perform the registration of the EBX module when the application server is launched. See Module registration [p 484].
`/META-INF/MANIFEST.MF`	Optional. If present, EBX reports the 'Implementation-Title' and 'Implementation-Version' values to *Administration > Configuration technique > Modules et modles de donnes*.
`/www/`	This optional directory contains all packaged resources, which are accessible via public URL. See Packaged resources [p 486].

Required files for Oracle WebLogic server:

`/WEB-INF/weblogic.xml`	WebLogic deployment descriptor file which activates the `prefer-web-inf-classes` policy, such as the following:
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<weblogic-web-app xmlns="http://xmlns.oracle.com/weblogic/weblogic-web-app">`
	`  <container-descriptor>`
	`  <prefer-web-inf-classes>true</prefer-web-inf-classes>`
	`  </container-descriptor>`
	`</weblogic-web-app>`
	See weblogic.xml Deployment Descriptor Elements for more information.

76.2 Module declaration

A module is declared using the document `/WEB-INF/ebx/module.xml`. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:ebx-schemas:module_2.4"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:ebx-schemas:module_2.4 http://schema.orchestranetworks.com/module_2.4.xsd">
 <name>moduleTest</name>
</module>
```

See the associated schema for documentation about each property. The main properties are as follows:

Element	Description	Required
`name`	Defines the unique identifier of the module in the server instance. The module name usually corresponds to the name of the web application (the name of its directory).	Yes.
`publicPath`	Defines a path other than the module's name identifying the web application in public URLs. This path is added to the URL of external resources of the module when computing absolute URLs. If this field is not defined, the public path is the module's `name`, defined above.	No.
`services`	Declares user services using the legacy API. See *Declaration and configuration* of legacy user services. From the version 5.8.0, it is strongly advised to use the new user services [p 587].	No.
`beans`	Declares reusable Java bean components. See the workflow package [p 573].	No.
`ajaxComponents`	Declares Ajax components. See **Declaring an Ajax component in a module** `UIAjaxComponent.declareInModule`[API] in the Java API.	No.

76.3 Module registration

In order to be identifiable by EBX, a module must be registered at runtime when the application server is launched. For a web application, every EBX module must:

- contain a Java class with the annotation `@WebListener` extending the class `ModuleRegistrationListener`[API].

> **Attention**
> When using the `@WebListener` annotation, ensure that the application server is configured to activate the servlet 3.0 annotation scanning for the web application. See JSR 315: JavaTM Servlet 3.0 Specification for more information.

or:

- contain a Servlet extending the class `ModuleRegistrationServlet`[API];
- make a standard declaration of this servlet in the deployment descriptor `/WEB-INF/web.xml`;
- ensure that this servlet will be registered at server startup by adding the following standard element to the deployment descriptor: `<load-on-startup>1</load-on-startup>`.

Additional recommendations and information:

- The method `handleRepositoryStartup` in `ModuleRegistrationServlet`[API] allows setting the logger associated with the module and defining additional behavior such as common JavaScript and CSS resources.
- The specific class extending `ModuleRegistrationServlet` must be located in the web application (under `/WEB-INF/classes` or `/WEB-INF/lib`; due to the fact that this class is internally used as a hook to the application's class-loader, to load Java classes used by the data models associated with the module).
- The application server startup process is asynchronous and web applications / EBX modules are discovered dynamically. The EBX repository initialization depends on this process and will wait for the registration of all used modules up to an unlimited amount of time. As a consequence, if a used module is not deployed for any reason, it must be declared in the EBX main configuration file. For more information, see the property Declaring modules as undeployed [p 385].
- All module registrations and unregistrations are logged in the `log.kernel` category.
- If an exception occurs while loading a module, the cause is written in the application server log.
- Once the servlet is out of service, the module is unregistered and the data models and associated datasets become unavailable. Note that hot deployment/undeployment is not supported [p 335].

Deployment descriptor example

Here is an example of a Java EE deployment descriptor (`/WEB-INF/web.xml`):

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                          http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
      version="3.0">
 <servlet>
    <servlet-name>InitEbxServlet</servlet-name>
    <servlet-class>com.foo.RegisterServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
 </servlet>
</web-app>
```

Registration example

Here is an implementation example of the `ModuleRegistrationServlet`:

```
package com.foo;
import javax.servlet.*;
import javax.servlet.http.*;
import com.onwbp.base.repository.*;
/**
  */
public class RegisterServlet extends ModuleRegistrationServlet
{

 public void handleRepositoryStartup(ModuleContextOnRepositoryStartup aContext)
  throws OperationException
 {
  // Perform module-specific initializations here
  ...

  // Declare custom resources here
  aContext.addExternalStyleSheetResource(MyCompanyResources.COMMON_STYLESHEET_URL);
  aContext.addExternalJavaScriptResource(MyCompanyResources.COMMON_JAVASCRIPT_URL);

  aContext.addPackagedStyleSheetResource("myModule.css");
  aContext.addPackagedJavaScriptResource("myModule.js");

 }

 public void handleRepositoryShutdown()
 {
  // Release resources of the current module when the repository is shut down here
  ...
 }

  public void destroyBeforeUnregisterModule()
 {
  // Perform operations when this servlet is being taken out of service here
  ...
 }

}
```

76.4 Packaged resources

The packaged resources are files and documents that can be directly accessed from client browsers and can be managed and specified either as `osd:resource` fields or via the Java API. They have various types and can also be localized.

Voir aussi

`ResourceType`[API]

Type `osd:resource` *[p 508]*

Directory structure

The packaged resources must be located under the following directory structure:

1. On the first level, the directory `/www/` must be located at the root of the module (web application).
2. On the second level, the directory must specify the localization. It can be:
   - `common/` should contain all the resources to be used by default, either because they are locale-independent or as the default localization (in EBX, the default localization is `en`, namely English);
   - `{lang}/` when localization is required for the resources located underneath, with `{lang}` to be replaced by the actual locale code; it should correspond to the locales supported by EBX; for more information, see Configuring EBX localization [p 373].

3. On the third level, the directory must specify the resource type. It can be:
   - `jscripts/` for JavaScript resources;
   - `stylesheets/` for Cascading Style Sheet (CSS) resources;
   - `html/` for HTML resources;
   - `icons/` for icon typed resources;
   - `images/` for image typed resources.

Example

In this example, the image `logoWithText.jpg` is the only resource that is localized:

```
/www
 ├── common
 │   ├── images
 │   │   ├── myCompanyLogo.jpg
 │   │   └── logoWithText.jpg
 │   ├── jscripts
 │   │   └── myCompanyCommon.js
 │   └── stylesheets
 │       └── myCompanyCommon.css
 ├── de
 │   └── images
 │       └── logoWithText.jpg
 └── fr
     └── images
         └── logoWithText.jpg
```

CHAPITRE 77

Mapping to Java

Ce chapitre contient les sections suivantes :

1. How to access data from Java?
2. Concurrency and isolation levels
3. Mapping of data types
4. Java bindings

77.1 How to access data from Java?

Read access

Data can be read from various generic Java classes, mainly `Adaptation`[API] and `ValueContext`[API]. The getter methods for these classes return objects that are typed according to the mapping rules described in the section Mapping of data types [p 490].

Write access

Data updates must be performed in a well-managed context:

- In the context of a procedure execution, by calling the methods `setValue...` of the interface `ValueContextForUpdate`[API], or
- During the user input validation, by calling the method `setNewValue` of the class `ValueContextForInputValidation`[API].

Modification of mutable objects

According to the mapping that is described in the Mapping of data types [p 490] section, some accessed Java objects are mutable objects. These are instances of `List` ,`Date` or any JavaBean. Consequently, these objects can be locally modified by their own methods. However, such modifications will remain local to the returned object unless one of the above setters is invoked and the current transaction is successfully committed.

77.2 Concurrency and isolation levels

Highest isolation level

The highest isolation level in ANSI/ISO SQL is `SERIALIZABLE`. Three execution methods guarantee the `SERIALIZABLE` isolation level within the scope of a dataspace:

- If the client code is run inside a `Procedure`[API] container. This is the case for every update, for exports to XML, CSV or archive, and for data services.
- If the client code accesses a dataspace that has been explicitly locked. See `LockSpec`[API].
- If the client code accesses data in a snapshot.

> **Note**
>
> For custom read-only transactions that run on a dataspace, it is recommended to use `ReadOnlyProcedure`[API].

Default isolation level

If the client code is run outside the contexts that enable `SERIALIZABLE`, its isolation level depends on the persistence mode:

- In semantic mode, the default isolation level is `READ UNCOMMITTED`.
- In relational mode, the default isolation level is the database default isolation level.

Voir aussi*Présentation des modes [p 267]*

Java access specificities

In a Java application, a record is represented by an instance of the `Adaptation` class. This object is initially linked to the corresponding persisted record. However, unless the client code is executed in a context that enables the SERIALIZABLE [p 490] isolation level, the object can become "disconnected" from the persisted record. If this occurs and concurrent updates have been performed, they will not be reflected in the `Adaptation` object.

Therefore, it is important for the client code to either be in a `SERIALIZABLE` context, or to regularly look up or refresh the `Adaptation` object.

Voir aussi

`AdaptationHome.findAdaptationOrNull`[API]

`AdaptationTable.lookupAdaptationByPrimaryKey`[API]

`Adaptation.getUpToDateInstance`[API]

77.3 Mapping of data types

This section describes how XML Schema type definitions and element declarations are mapped to Java types.

Simple data types

Basic rules for simple data types

Each XML Schema simple type corresponds to a Java class, the mapping is documented in the table XML Schema built-in simple types [p 504].

Voir aussi *`SchemaNode.createNewOccurrence`*[API]

Multiple cardinality on a simple element

If the attribute `maxOccurs` is greater than 1, the element is an aggregated list and the corresponding instance in Java is an instance of `java.util.List`.

Elements of the list are instances of the Java class that is determined from the mapping of the simple type (see previous section).

Complex data types

Complex type definitions without a class declaration

By default (no attribute `osd:class`), a terminal node of a complex type is instantiated using an internal class. This class provides a generic JavaBean implementation. However, if a custom client Java code has to access these values, it is recommended to use a custom JavaBean. To do so, use the `osd:class` declaration described in the next section.

It is also possible to transparently instantiate, read and modify the mapped Java object, with or without the attribute `osd:class`, by invoking the methods `SchemaNode.createNewOccurrence`[API], `SchemaNode.executeRead`[API] and `SchemaNode.executeWrite`[API].

Mapping of complex types to custom JavaBeans

It is possible to map an XML Schema complex type to a custom Java class. This is done by adding the attribute `osd:class` to the complex node definition. Unless the element has `xs:maxOccurs > 1`, you must also specify the attribute `osd:access` for the node to be considered a *terminal* node. If the element has `xs:maxOccurs > 1`, it is automatically considered to be terminal.

The custom Java class must conform to the JavaBean protocol. This means that each child of the complex type must correspond to a JavaBean property of the class. Additionally, each JavaBean property must be a read-write property, and its implementation must ensure that the value set by the setter method is returned, as-is, by the getter method. Contextual computations are not allowed in these methods.

Example

In this example, the Java class `com.carRental.Customer` must define the methods `getFirstName()` and `setFirstName(String)`.

A JavaBean can have a custom user interface within TIBCO EBX, by using a `UIBeanEditor`[API].

```
<xs:element name="customer" osd:access="RW">
  <xs:complexType name="subscriber" osd:class="com.carRental.Customer">
 <xs:sequence>
   <xs:element name="firstName" type="xs:string"/>
   ...
 </xs:sequence>
  </xs:complexType>
</xs:element>
```

Multiple cardinality on a complex element

If the attribute `maxOccurs` is greater than 1, then the corresponding instance in Java is:

- An instance of `java.util.List` for an aggregated list, where every element in the list is an instance of the Java class determined by the mapping of simple types [p 504], or
- An instance of `AdaptationTable`[API], if the property `osd:table` is specified.

77.4 Java bindings

Java bindings allow generating Java types that reflect the structure of the data model. The Java code generation can be done in the user interface. See Generating Java bindings [p 495].

Benefits

Ensuring the link between XML Schema structure and Java code provides a number of benefits:

- **Development assistance:** Auto-completion when typing an access path to parameters, if supported by your IDE.
- **Access code verification:** All accesses to parameters are verified at code compilation.
- **Impact verification:** Each modification of the data model impacts the code compilation state.
- **Cross-referencing:** By using the reference tools of your IDE, it is easy to verify where a parameter is used.

Consequently, it is strongly recommended to use Java bindings.

XML declaration

The specification of the Java types to be generated from the data model is included in the main schema.

Each binding element defines a generation target. It must be located at, in XPath notation, `xs:schema/xs:annotation/xs:appinfo/ebxbnd:binding`, where the prefix `ebxbnd` is a reference to the namespace identified by the URI `urn:ebx-schemas:binding_1.0`. Several binding elements can be defined if you have different generation targets.

The attribute `targetDirectory` of the element `ebxbnd:binding` defines the root directory used for Java type generation. Generally, it is the directory containing the project source code, `src`. A relative path is interpreted based on the current runtime directory of the VM, as opposed to the XML schema.

See bindings XML Schema.

XML bindings example

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:ebxbnd="urn:ebx-schemas:binding_1.0">
 <xs:annotation>
  <xs:appinfo>
   <!-- The bindings define how this schema will be represented in Java.
   Several <binding> elements may be defined, one for each target. -->
   <ebxbnd:binding
    targetDirectory="../_ebx-demos/src-creditOnLineStruts-1.0/">
    <javaPathConstants typeName="com.creditonline.RulesPaths">
     <nodes root="/rules" prefix="" />
    </javaPathConstants>
    <javaPathConstants typeName="com.creditonline.StylesheetConstants">
     <nodes root="/stylesheet" prefix="" />
    </javaPathConstants>
   </ebxbnd:binding>
  </xs:appinfo>
 </xs:annotation>
 ...
```

```
</xs:schema>
```

Java constants can be defined for XML schema paths. To do so, generate one or more interfaces from a schema node, including the root node `/`. The example generates two Java path constant interfaces, one from the node `/rules` and the other from the node `/stylesheet` in the schema. Interface names are described by the element `javaPathConstants` with the attribute `typeName`. The associated node is described by the element `nodes` with the attribute `root`.

CHAPITRE 78

Tools for Java developers

TIBCO EBX provides Java developers with tools to facilitate use of the EBX API, as well as integration with development environments.

Ce chapitre contient les sections suivantes :

1. Activating the development tools
2. Data model refresh tool
3. Generating Java bindings
4. Path to a node
5. Web component link generator

78.1 Activating the development tools

To activate the development tools, run EBX in *development mode*. This is specified in the EBX main configuration file EBX run mode [p 387] using the property `backend.mode=development`.

78.2 Data model refresh tool

When editing the data model directly as an XML Schema document without using the data-modeling tool provided by EBX, you can refresh it without restarting the application server.

In the 'Administration' area, select **Select > Configuration technique > Outils de dveloppement > Synchroniser les schmas mis jour** (or **Synchroniser tous les schmas**).

> **Attention**
> Since the operation is critical regarding data consistency, refreshing the data models acquires a global exclusive lock on the repository. This means that most other operations (data access and update, validation, etc.) will wait until the completion of the data model refresh.

78.3 Generating Java bindings

The Java types specified by Java bindings can be generated from a dataset or a data model, by selecting **Actions > Generate Java** in the navigation pane.

Voir aussi*Java bindings [p 492]*

78.4 Path to a node

The field 'Data path' is displayed in the documentation pane of a node. This field indicates the path to the node, which can be useful when writing XPath formulas.

> **Note**
>
> This field is always available to administrators.

78.5 Web component link generator

The 'Web component link generator' service is a user interface designed to create HTTP requests that call EBX web components. To launch this service, select **Actions > Web component link generator** in the navigation pane.

CHAPITRE 79

Terminology changes

A new TIBCO EBX release can introduce new vocabulary for users. To preserve the backward compatibility, these terminology changes do not usually impact the API. Consequently, Java class names, method names, data services operation names, etc. still use the older version terminology. This chapter purpose is to facilitate the correspondence of the old term in the API to the new terms.

Voir aussi*Glossaire* *[p 23]*

Ce chapitre contient les sections suivantes :

1. Terminology changes in version 5.9
2. Terminology changes in version 5.0

79.1 Terminology changes in version 5.9

New term	Term prior to version 5.9.0
D3 primary node	D3 master node
D3 replica node	D3 slave node

79.2 Terminology changes in version 5.0

The following table summarizes the mappings between the version 5.0.0 terminology and previous terminology:

New term	Term prior to version 5.0.0
Dataset	Adaptation instance
Child dataset	Child adaptation instance
Data model	Data model
Dataspace	Branch
Snapshot	Version
Dataspace or snapshot	Home
Data Workflow	Workflow instance
Workflow model	Workflow definition
Workflow publication	Workflow
Data services	Data services
Field	Attribute
Inherited field	Inherited attribute
Record	Record/occurrence
Validation rule	Constraint
Simple/advanced control	Simple/advanced constraint

Data model

CHAPITRE 80

Introduction

A data model is a structural definition of the data to be managed in the TIBCO EBX repository. Data models contribute to EBX's ability to guarantee the highest level of data consistency and to facilitate data management.

Specifically, the data model is a document that conforms to the XML Schema standard (W3C recommendation). Its main features are as follows:

- A rich library of well-defined simple data types [p 503], such as integer, boolean, decimal, date, time;
- The ability to define additional simple types [p 505] and complex types [p 505];
- The ability to define simple lists of items, called aggregated lists [p 514];
- Validation constraints [p 537] (facets), for example: enumerations, uniqueness constraints, minimum/maximum boundaries.

EBX also uses the extensibility features of XML Schema for other useful information, such as:

- Predefined types [p 506], for example: locale, resource, html;
- Definition of tables [p 517] and foreign key constraints [p 522];
- Mapping data in EBX to Java beans;
- Advanced validation constraints [p 537] (extended facets), such as dynamic enumerations;
- Extensive presentation information [p 555], such as labels, descriptions, and error messages.

> **Note**
>
> EBX supports a subset of the W3C recommendations, as some features are not relevant to Master Data Management.

Ce chapitre contient les sections suivantes :

1. Editing the data model
2. References
3. Relationship between datasets and data models
4. Pre-requisite for XML Schemas
5. Conventions
6. Schemas with reserved names

80.1 Editing the data model

There are two different ways to define a data model:

- The data model can be defined using an XML Schema editor or through the data model assistant. The data model assistant has the advantage of being integrated into the EBX user interface, abstracting the verbose underlying XML. For more information, see Introduction aux modles de donnes [p 34]. The data model assistant allows using features that are not documented to be used outside of the DMA; e.g. Toolbars and Widgets.
- By using an external XML Schema document editor.

80.2 References

For an introduction to XML Schema, see the W3Schools XML Schema Tutorial.

Voir aussi

XML Schema Part 0: Primer

XML Schema Part 1: Structures

XML Schema Part 2: Datatypes

80.3 Relationship between datasets and data models

Each root dataset is associated with a single data model. At the dataspace creation, an associated data model is selected, on which to base the dataset.

Voir aussi*Cration du jeu de donnes [p 123]*

80.4 Pre-requisite for XML Schemas

In order for an XML Schema to be accepted by EBX, it must include a global element declaration that includes the attribute `osd:access="--"`.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:osd="urn:ebx-schemas:common_1.0" xmlns:fmt="urn:ebx-schemas:format_1.0">
   <xs:import namespace="urn:ebx-schemas:common_1.0"
     schemaLocation="http://schema.orchestranetworks.com/common_1.0.xsd"/>
   <xs:element name="root" osd:access="--">
  ...
  </xs:element>
</xs:schema>
```

80.5 Conventions

By convention, namespaces are always defined as follows:

Prefix	Namespace
`xs:`	`http://www.w3.org/2001/XMLSchema`
`osd:`	`urn:ebx-schemas:common_1.0`
`fmt:`	`urn:ebx-schemas:format_1.0`
`usd:`	`urn:ebx-schemas:userServices_1.0`
`emd:`	`urn:ebx-schemas:entityMappings_1.0`

80.6 Schemas with reserved names

Several data models in EBX have reserved names.

All references to other data models (using the attribute `schemaLocation` for an import, include or redefine) that end with one of the following strings are reserved:

- common_1.0.xsd
- org_1.0.xsd
- coreModel_1.0.xsd
- session_1.0.xsd
- userServices_1.0.xsd
- entityMappings_1.0.xsd

These XSD files correspond to the schemas provided for the module `ebx-root-1.0`, at the path `/WEB-INF/ebx/schemas`. The attribute `schemaLocation` can reference the files at this location or a copy, if the file names are identical. This is useful if you want to avoid a module dependency on `ebx-root-1.0`.

For security reasons, EBX uses an internal definition for these schemas to prevent any modification.

CHAPITRE 81

Data types

This chapter details the data types supported by TIBCO EBX.

Voir aussi*Tables and relationships* *[p 517]*

Ce chapitre contient les sections suivantes :

1. XML Schema built-in simple types
2. XML Schema named simple types
3. XML Schema complex types
4. Extended simple types defined by EBX
5. Complex types defined by EBX
6. Aggregated lists
7. Including external data models

81.1 XML Schema built-in simple types

The table below lists all the simple types defined in XML Schema that are supported by EBX, along with their corresponding Java types.

XML Schema type	Java class	Notes
xs:string	java.lang.String	
xs:boolean	java.lang.Boolean	
xs:decimal	java.math.BigDecimal	A totalDigits facet with a value equal to 15 is added by default to decimal fields that are contained in a mapped table (relational, historized or replicated table). However, this facet can be overwritten with a greater value in the data model.
xs:dateTime	java.util.Date	
xs:time	java.util.Date	The date portion of the returned Date is always set to '1970/01/01'.
xs:date	java.util.Date	The time portion of the returned Date is always the beginning of the day, that is, '00:00:00'.
xs:anyURI	java.net.URI	
xs:Name (xs:string restriction)	java.lang.String	
xs:int (xs:decimal restriction)	java.lang.Integer	
xs:integer (xs:decimal restriction)	java.lang.Integer	This mapping does not comply with the XML Schema recommendation. Although the XML Schema specification states that xs:integer has no value space limitation, this value space is, in fact, restricted by the Java specifications of the java.lang.Integer object.

The mapping between XML Schema types and Java types are detailed in the section Mapping of data types [p 490].

81.2 XML Schema named simple types

Named simple types can be defined when designing a data model for redefining an existing built-in simple type. A *named simple type* can be reused in the data model.

Restrictions:

- In the data model, only the element `restriction` is allowed in a named simple type, and even then, only derivation by restriction is supported. Notably, the elements `list` and `union` are not supported.
- Facet definition is not cumulative. That is, if an element and its named type both define the same kind of facet, then the facet defined in the type is overridden by the local facet definition. However, this restriction does not apply to programmatic facets defined by the element `osd:constraint`. For `osd:constraint`, if an element and its named type both define a programmatic facet with different Java classes, the definition of these facets will be cumulative. Contrary to the XML Schema Specification, EBX is not strict regarding the definition of a facet of the same kind in an element and its named type. That is, the value of a same kind of facet defined in an element is not checked according to the one defined in the named type. However, in the case of static enumerations defined both in an element and its type, the local enumeration will be replaced by the intersection between these enumerations.
- It is not possible to define different types of enumerations on both an element and its named type. For instance, you cannot specify a static enumeration in an element and a dynamic enumeration in its named type.
- It is not possible to simultaneously define a pattern facet in both an element and its named type.

81.3 XML Schema complex types

Complex types can be defined when designing a data model. A *named complex type* can be reused in the data model.

Restrictions:

- In the data model, only the element `sequence` is allowed. Notably, attribute definition is not supported.
- Type extensions are not supported in the current version of EBX.

81.4 Extended simple types defined by EBX

EBX provides pre-defined simple data types:

XML Schema type	Java class
`osd:text` (`xs:string` restriction)	`java.lang.String`
`osd:html` (`xs:string` restriction)	`java.lang.String`
`osd:email` (`xs:string` restriction)	`java.lang.String`
`osd:password` (`xs:string` restriction)	`java.lang.String`
`osd:color` (`xs:string` restriction)	`java.lang.String`
`osd:resource` (`xs:anyURI` restriction)	internal class
`osd:locale` (`xs:string` restriction)	`java.util.Locale`
`osd:dataspaceKey` (`xs:string` restriction)	`java.lang.String`
`osd:datasetName` (`xs:string` restriction)	`java.lang.String`

The above types are defined by the internal schema `common-1.0.xsd`. They are defined as follows:

`osd:text`	This type represents textual information. For a basic `xs:string`, its default user interface in EBX consists of a dedicated editor with several lines for input and display.
	`<xs:simpleType name="text">`
	` <xs:restriction base="xs:string" />`
	`</xs:simpleType>`

`osd:html`	This represents a character string with HTML formatting. A WYSIWYG editor is provided in EBX.
	`<xs:simpleType name="html">`
	` <xs:restriction base="xs:string" />`
	`</xs:simpleType>`

`osd:email`	This represents an email address as specified by the RFC822 standard.
	`<xs:simpleType name="email">`
	` <xs:restriction base="xs:string" />`
	`</xs:simpleType>`

`osd:password`	This represents a hashed or encrypted password. A specific editor is provided in EBX.
	`<xs:element name="password" type="osd:password" />`
	The default editor performs a hash computation using the SHA-512 algorithm. This encryption function is also available from a Java client using the method `DirectoryDefault.encryptString`[API].
	It is also possible for the default editor to use a different encryption mechanism by specifying a class that implements the interface `Encryption`[API].
	`<xs:element name="password" type="osd:password">`
	` <xs:annotation>`
	`  <xs:appinfo>`
	`  <osd:uiBean class="com.orchestranetworks.ui.UIPassword">`
	`  <encryptionClass>package.EncryptionClassName</encryptionClass>`
	`  </osd:uiBean>`
	`  </xs:appinfo>`
	` </xs:annotation>`
	`</xs:element>`
	It is possible to specify some salt by referencing a path to another field, and by using a class the implements the interface `HashComputation`[API].
	`<xs:element name="password" type="osd:password">`
	` <xs:annotation>`
	`  <xs:appinfo>`
	`  <osd:uiBean class="com.orchestranetworks.ui.UIPassword">`
	`  <encryptionClass>package.HashClassName</encryptionClass>`
	`  <saltPath>../login</saltPath>`
	`  </osd:uiBean>`
	`  </xs:appinfo>`
	` </xs:annotation>`
	`</xs:element>`

`osd:locale`

This represents a geographical, political or cultural location. The locale type is translated into Java by the class `java.util.Locale`.

```
<xs:simpleType name="locale">
 <xs:restriction base="xs:string">
  <xs:enumeration value="ar" osd:label="Arabic" />
  <xs:enumeration value="ar_AE" osd:label="Arabic (United Arab
 Emirates)" />
  <xs:enumeration value="ar_BH" osd:label="Arabic (Bahrain)" />
  <xs:enumeration value="ar_DZ" osd:label="Arabic (Algeria)" />
  <xs:enumeration value="ar_EG" osd:label="Arabic (Egypt)" />
  <xs:enumeration value="ar_IQ" osd:label="Arabic (Iraq)" />
   ...
  <xs:enumeration value="vi_VN" osd:label="Vietnamese (Vietnam)" /
>
  <xs:enumeration value="zh" osd:label="Chinese" />
  <xs:enumeration value="zh_CN" osd:label="Chinese (China)" />
  <xs:enumeration value="zh_HK" osd:label="Chinese (Hong Kong)" />
  <xs:enumeration value="zh_TW" osd:label="Chinese (Taiwan)" />
 </xs:restriction>
</xs:simpleType>
```

`osd:color`

This represents a character string with hexadecimal RGB color formatting. A color picker `UIComponent` is provided in EBX.

```
<xs:simpleType name="color">
 <xs:restriction base="xs:string" />
</xs:simpleType>
```

`osd:resource`

This represents a resource packaged in a module. For more information, see Packaged resources [p 486]. This type requires the definition of the facet FacetOResource [p 542].

```
<xs:simpleType name="resource">
 <xs:restriction base="xs:anyURI" />
</xs:simpleType>
```

`osd:dataspaceKey`

This type represents a reference to a dataspace.

```
<xs:element name="dataspaceField" type="osd:dataspaceKey" />
```

A specific editor is provided in EBX that displays the dataspaces that can be referenced.

It is possible to specify the dataspaces that can be referenced using the element `osd:dataspaceSet` under `xs:annotation/xs:appInfo`. If the element `osd:dataspaceSet` is not defined, then by default, only open branches can be referenced.

```
<xs:element name="dataspaceField" type="osd:dataspaceKey">
 <xs:annotation>
  <xs:appinfo>
   <osd:dataspaceSet>
    <include>
     <pattern>a pattern</pattern>
     <type>all | branch | version</type>
     <includeDescendants>none | allDescendants |
 allBranchDescendants | allSnapshotDescendants | branchChildren |
 snapshotChildren</includeDescendants>
    </include>
    <exclude>
     <pattern>a pattern</pattern>
     <type>all | branch | version</type>
```

```
      <includeDescendants>none | allDescendants |
  allBranchDescendants | allSnapshotDescendants | branchChildren |
  snapshotChildren</includeDescendants>
     </include>
     <filter osd:class="com.foo.MyDataspaceFilter">
      <param1>...</param1>
      <param2>...</param2>
     </filter>
    </osd:dataspaceSet>
   </xs:appinfo>
  </xs:annotation>
 </xs:element>
```

- includes

  Dfinit les espaces de donnes qui peuvent tre rfrencs par ce champ. An `include` must at least be defined.

  `pattern`: Dfinit une expression rgulire laquelle le nom d'un espace de donnes doit tre conforme. This property is mandatory.

  `type`: Dfinit le type d'espaces de donnes (branches ou versions) qui peuvent tre rfrencs par ce champ. Si non dfinie, cette restriction s'appliquera aux espaces de donnes de type branche. If `all` then branches and snapshots are included. If `branch` then only branches are included. If `snapshot` then only snapshots are included. If not set, this property is `branch` by default.

  `includeDescendants`: Specifies if children or descendants of the dataspaces that match the specified pattern are included in the set. If `none` then neither children nor descendants of the dataspaces that match the specified pattern are included. If `allDescendants` then all descendants of the dataspaces that match the specified pattern are included. If `allBranchDescendants` then all descendant branches of the dataspaces that match the specified pattern are included. If `allSnapshotDescendants` then all descendant snapshots of the dataspaces that match the specified pattern are included. If `directBranchChildren` then only direct branches of the dataspaces that match the specified pattern are included. If `directSnapshotChildren` then only direct snapshots of the dataspaces that match the specified pattern are included. If not set, this property is `none` by default.

- excludes

  Dfinit les espaces de donnes qui ne peuvent pas tre rfrencs par ce champ. Les exclusions sont ignores si la configuration ne dfinit pas d'inclusion.

  `pattern`: Dfinit une expression rgulire laquelle le nom d'un espace de donnes doit tre conforme. This property is mandatory.

  `type`: Dfinit le type d'espaces de donnes (branches ou versions) qui peuvent tre rfrencs par ce champ. Si

non dfinie, cette restriction s'appliquera aux espaces de donnes de type branche. If `all` then branches and snapshots are excluded. If `branch` then only branches are excluded. If `snapshot` then only snapshots are excluded. If not set, this property is `branch` by default.

`includeDescendants:` Specifies if children or descendants of the datasets that match the specified pattern are excluded from the set. If `none` then neither children nor descendants of the dataspaces that match the specified pattern are excluded. If `allDescendants` then all descendants of the dataspaces that match the specified pattern are excluded. If `allBranchDescendants` then all descendant branches of the dataspaces that match the specified pattern are excluded. If `allSnapshotDescendants` then all descendant snapshots of the dataspaces that match the specified pattern are excluded. If `directBranchChildren` then only direct branches of the dataspaces that match the specified pattern are excluded. If `directSnapshotChildren` then only direct snapshots of the dataspaces that match the specified pattern are excluded. If not set, this property is `none` by default.

- `filter`

  Dfinit un filtre pour accepter ou refuser des espaces de donnes dans le contexte d'un jeu de donnes ou d'un enregistrement. Ce filtre est uniquement utilis par le composant de saisie ddi ce type de champ. Ce filtre n'est pas utilis lors de la validation de ce champ. Des contrles additionnels peuvent tre dfinis en utilisant une contrainte spcifique (composant).

  The attribute `osd:class` specifies a Java bean that implements the interface `DataspaceSetFilter`[API].

It is also possible to customize validation messages and the control policy associated with this type using the element `validation` under `xs:annotation/xs:appInfo/osd:dataspaceSet`. See Facet validation message with severity [p 558] and Control policy [p 546] for more information.

`osd:datasetName`

This type represents a reference to a dataset.

```
<xs:element name="dataset" type="osd:datasetName" />
```

A specific editor provided in EBX displays the datasets that can be referenced.

It is also possible to specify the datasets that can be referenced using the element `osd:datasetSet` under `xs:annotation/xs:appInfo`:

```
<xs:element name="datasetField" type="osd:datasetName">
 <xs:annotation>
  <xs:appinfo>
   <osd:datasetSet>
    <branch>productsBranch</branch>
    <version>productsVersion</version>
    <dataspaceSelector>../dataspaceField</dataspaceSelector>
    <pattern>a pattern</pattern>
    <filter osd:class="com.foo.MyDatasetFilter">
     <param1>...</param1>
     <param2>...</param2>
    </filter>
   </osd:datasetSet>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

- `branch`

  Spcifie la branche source. Seuls les jeux de donnes contenus dans cette branche pourront tre rfrencs par un champ de type Identifiant de jeux de donnes (osd:datasetName).

- `version`

  Spcifie la version source. Seuls les jeux de donnes contenus dans cette version pourront tre rfrencs par un champ de type Identifiant de jeux de donnes (osd:datasetName).

- `dataspaceSelector`

  Specifies a field in the same data model that defines the dataspace containing the datasets that can be referenced. The specified field must be of type `xs:string` or `osd:dataspaceKey`. The value of this field must comply with the representation of a persistent identifier of a dataspace or snapshot. See `HomeKey.format`[API] for more information.

  The referred node must respect the restrictions existing for dynamic facets, see Dynamic constraints [p 540].

- `includes`

  Dfinit les jeux de donnes qui peuvent tre rfrencs par ce champ.

  `pattern`: Dfinit une expression rgulire laquelle le nom d'un jeu de donnes doit tre conforme. This property is mandatory.

  `includeDescendants`: Dfinit si les jeux de donnes enfants ou descendants sont inclus dans cet ensemble. Si non dfinie, cette restriction ne s'appliquera pas aux jeux de donnes enfants. If `none` then neither children nor descendants of the datasets that match the specified pattern are excluded. If `directChildren` then only direct children of the datasets that match the specified pattern are excluded. If `allDescendants` then all descendants of the datasets that match the specified

pattern are excluded. If not set, this property is `none` by default.

- `excludes`

  Dfinit les jeux de donnes qui ne peuvent pas tre rfrencs par ce champ. Les exclusions sont ignores si la configuration ne dfinit pas d'inclusion.

  `pattern`: Dfinit une expression rgulire laquelle le nom d'un jeu de donnes doit tre conforme. This property is mandatory.

  `includeDescendants`: Dfinit si les jeux de donnes enfants ou descendants sont inclus dans cet ensemble. Si non dfinie, cette restriction ne s'appliquera pas aux jeux de donnes enfants. If `none` then neither children nor descendants of the datasets that match the specified pattern are excluded. If `directChildren` then only direct children of the datasets that match the specified pattern are excluded. If `allDescendants` then all descendants of the datasets that match the specified pattern are excluded. If not set, this property is `none` by default.

- `filter`

  Dfinit un filtre pour accepter ou refuser des jeux de donnes dans le contexte d'un jeu de donnes ou d'un enregistrement. Ce filtre est uniquement utilis par le composant de saisie ddi ce type de champ. Ce filtre n'est pas utilis lors de la validation de ce champ. Des contrles additionnels peuvent tre dfinis en utilisant une contrainte spcifique (composant).

  The attribute `osd:class` specifies a Java bean that implements the interface `DatasetSetFilter`[API]. A validation message is added to the associated field if an input dataspace reference does not match this filter.

One of the elements `branch`, `version` or `dataspaceSelector` must be defined.

It is also possible to customize validation messages and the control policy associated with this type using the element `validation` under `xs:annotation/xs:appInfo/osd:datasetSet`. See Facet validation message with severity [p 558] and Control policy [p 546] for more information.

81.5 Complex types defined by EBX

EBX provides pre-defined complex data types:

XML Schema type	Description
`osd:UDA`	**User Defined Attribute:** This type allows any user, according to their access rights, to define a value associated with an attribute defined in a dictionary called a UDA Catalog.
`osd:UDACatalog`	**Catalog of User Defined Attributes:** This type consists of a table in which attributes can be specified. This catalog is used by all `osd:UDA` elements declared in the same data model.

`osd:UDA`

A User Defined Attribute (UDA) supports both the `minOccurs` and `maxOccurs` attributes, as well as the attribute `osd:UDACatalogPath`, which specifies the path of the corresponding catalog.

```
<xs:element name="firstUDA" type="osd:UDA" minOccurs="0"
 maxOccurs="unbounded" osd:UDACatalogPath="//insuranceCatalog" />
<xs:element name="secondUDA" type="osd:UDA" minOccurs="1"
 maxOccurs="1"
 osd:UDACatalogPath="/root/userCatalog" />
<xs:element name="thirdUDA" type="osd:UDA" minOccurs="0"
 maxOccurs="1"
 osd:UDACatalogPath="//userCatalog" />
```

In the manager, when working with a UDA, the editor will adapt itself to the type of the selected attribute.

`osd:UDACatalog`

Internally, a catalog is represented as a table. The parameters `minOccurs` and `maxOccurs` must be specified.

Several catalogs can be defined in the same data model.

```
<xs:element name="insuranceCatalog" type="osd:UDACatalog"
 minOccurs="0" maxOccurs="unbounded">
 <xs:annotation>
  <xs:documentation xml:lang="en-US">Insurance Catalog.</
xs:documentation>
  <xs:documentation xml:lang="fr-FR">Catalog assurance.</
xs:documentation>
 </xs:annotation>
</xs:element>
<xs:element name="userCatalog" type="osd:UDACatalog" minOccurs="0"
 maxOccurs="unbounded">
 <xs:annotation>
  <xs:documentation xml:lang="en-US">User catalog.</
xs:documentation>
  <xs:documentation xml:lang="fr-FR">Catalogue utilisateur.</
xs:documentation>
 </xs:annotation>
</xs:element>
```

Only the following types are available for creating new attributes:

- xs:string
- xs:boolean
- xs:decimal

- xs:dateTime
- xs:time
- xs:date
- xs:anyURI
- xs:Name
- xs:int
- osd:html
- osd:email
- osd:password
- osd:locale
- osd:text

Restrictions on User Defined Attributes and Catalogs

The following features are unsupported on UDA elements:

- Facets
- Functions using the `osd:function` property
- UI bean editors using the `osd:uiBean` property
- The `osd:checkNullInput` property
- History features
- Replication
- Inheritance features, using the `osd:inheritance` property

As UDA catalogs are internally considered to be tables, the restrictions that apply to tables also exist for `UDACatalog` elements.

81.6 **Aggregated lists**

In XML Schema, the maximum number of times an element can occur is determined by the value of the `maxOccurs` attribute in its declaration. If this value is strictly greater than 1 or is `unbounded`, the data can have multiple occurrences. If no `osd:table` declaration is included, this element is called an *aggregated list*. In Java, it is then represented as an instance of the class `java.util.List`.

The following is an example of an aggregated list that defines the pricing of a loan product, depending on the amount borrowed.

```
<xs:element name="pricing" minOccurs="0" maxOccurs="unbounded"
 osd:access="RW">
 <xs:annotation>
  <xs:documentation>
   <osd:label>Pricing</osd:label>
   <osd:description>Pricing grid </osd:description>
  </xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:sequence>
   <xs:element name="amount" type="xs:int">
    <xs:annotation>
     <xs:documentation>
      <osd:label>Amount borrowed</osd:label>
     </xs:documentation>
```

```
    </xs:annotation>
   </xs:element>
   <xs:element name="monthly" type="xs:int">
    <xs:annotation>
     <xs:documentation>
      <osd:label>Monthly payment </osd:label>
     </xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="cost" type="xs:int">
    <xs:annotation>
     <xs:documentation>
      <osd:label>Cost</osd:label>
     </xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Aggregated lists have a dedicated editor in EBX. This editor allows you to add or to delete occurrences.

Attention

The addition of an `osd:table` declaration to an element with `maxOccurs > 1` is a very important consideration that must be taken into account during the design process. An aggregated list is severely limited with respect to the many features that are supported by tables. Some features unsupported on aggregated lists that are supported on tables are:

- Performance and memory optimization;
- Lookups, filters and searches;
- Sorting, view and display in hierarchies;
- Identity constraints (primary keys and uniqueness constraints);
- Detailed permissions for creation, modification, deletion and particular permissions at the record level;
- Detailed comparison and merge.

Thus, *aggregated lists should be used only for small volumes of simple data (one or two dozen occurrences), with no advanced requirements*. For larger volumes of data or more advanced functionalities, it is strongly advised to use an `osd:table` declaration.

For more information on table declarations, see Tables and relationships [p 517].

81.7 Including external data models

Including another data model in your current model allows you to use the reusable types that are defined in that data model. You can thus use the inclusion of external data models to share data types between multiple XML Schema Documents.

To include another XML Schema Document in your model, thereby including the data types that it defines, specify the `xs:include` element as follows:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:osd="urn:ebx-schemas:common_1.0" xmlns:fmt="urn:ebx-schemas:format_1.0">
   <xs:include  schemaLocation="./schemaToInclude.xsd"/>
 ...
</xs:schema>
```

The attribute `schemaLocation` is mandatory and must specify either an absolute or a relative path to the XML Schema Document to include.

The inclusion of XML Schema Documents is not namespace aware, thus all included data types must belong to the same namespace. As a consequence, including XML Schema Documents that define data types of the same name is not supported.

EBX includes extensions with specific URNs for including embedded data models and data models packaged in modules.

To include an embedded data model in a model, specify the URN defined by EBX. For example:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:osd="urn:ebx-schemas:common_1.0" xmlns:fmt="urn:ebx-schemas:format_1.0">
 <xs:include schemaLocation="urn:ebx:publication:myPublication"/>
 ...
</xs:schema>
```

To include a data model packaged in a module, specify the specific URN defined by EBX. For example:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:osd="urn:ebx-schemas:common_1.0" xmlns:fmt="urn:ebx-schemas:format_1.0">
 <xs:include schemaLocation="urn:ebx:module:aModuleName:/WEB-INF/ebx/schema/myDataModel.xsd"/>
 ...
</xs:schema>
```

See `SchemaLocation`[API] for more information about specific URNs supported by EBX.

> **Note**
>
> If the packaged data model uses Java resources, the class loader of the module containing the data model will be used at runtime for resolving these resources.

CHAPITRE **82**

Tables and relationships

Ce chapitre contient les sections suivantes :

1. Tables
2. Foreign keys
3. Associations

82.1 Tables

Overview

TIBCO EBX supports the features of relational database tables, including the handling of large volumes of records, and identification by primary key.

Tables provide many benefits that are not offered by aggregated lists [p 514]. Beyond relational capabilities, some features that tables provide are:

- filters and searches;
- sorting, views and hierarchies;
- identity constraints: primary keys, foreign keys [p 522] and uniqueness constraints [p 539];
- specific permissions for creation, modification, and deletion;
- dynamic and contextual permissions at the individual record level;
- detailed comparison and merge;
- ability to have inheritance at the record level (see dataset inheritance [p 296]);
- performance and memory optimization.

Voir aussi

Foreign keys [p 522]

Associations [p 525]

Actions sur les jeux de donnes existants [p 139]

Vue tabulaire simple [p 129]

Vues hirarchiques [p 129]

Historique [p 273]

Declaration

A table element, which is an element with *maxOccurs > 1*, is declared by adding the following annotation:

```
<xs:annotation>
 <xs:appinfo>
  <osd:table>
   <primaryKeys>/pathToField1 /pathToField...n</primaryKeys>
  </osd:table>
 </xs:appinfo>
</xs:annotation>
```

Common properties

Element	Description	Required
`primaryKeys`	Specifies the primary key fields of the table. Each field of the primary key must be denoted by its absolute XPath notation that starts just under the root element of the table. If there are multiple fields in the primary key, the list is delimited by whitespace. **Note:** Whitespaces in primary keys of type `xs:string` are handled differently. See Whitespace handling for primary keys of type string [p 550].	Yes.

`defaultLabel`	Defines the end-user display of records. Multiple variants can be specified:
	• A static non-localized expression is defined using the `defaultLabel` element, for example:
	`<defaultLabel>Product: ${./productCode}</defaultLabel>`
	• Static localized expressions are specified using the `defaultLabel` element with the attribute `xml:lang`, for example:
	`<defaultLabel xml:lang="fr-FR">Produit : ${./productCode}</defaultLabel>`
	`<defaultLabel xml:lang="en-US">Product: ${./productCode}</defaultLabel>`
	• A JavaBean that implements the interface `UILabelRenderer`[API] and/or the interface `UILabelRendererForHierarchy`[API]. The JavaBean is specified by means of the attribute `osd:class`, for example:
	`<defaultLabel osd:class="com.wombat.MyLabel"></defaultLabel>`
	Note: The priority of the tags when displaying the user interface is the following:
	1. `defaultLabel` tags with a JavaBean (but it is not allowed to define several renderers of the same type);
	2. `defaultLabel` tags with a static localized expression using the `xml:lang` attribute;
	3. `defaultLabel` tags with a static non-localized expression.
	Attention: Les droits d'accs dfinis dans les jeux de donnes associs ne sont pas appliqus lors de l'affichage des libells. Les champs habituellement cachs cause de droits d'accs seront affichs dans les libells des enregistrements.
		No.

`index`	Specifies an index for speeding up requests that match this index (see performances [p 319]). The attribute `name` is mandatory. Each field of the index must be denoted by its absolute XPath notation, which starts just under the root element of the table. If there are multiple fields in the index, the list is delimited by whitespace. **Note:** • Indexing only concerns semantic and relational tables. History and replica tables are not affected. • It is possible to define multiple indexes on a table. • It is not possible to define two indexes with the same name.	No.

Element	Description	Required
	• It is not possible to declare two indexes containing the exact same fields. • An indexed field must be terminal. • An indexed field cannot be a list nor under a list. • A field declared as an *inherited field* cannot be indexed. • A field declared as a function cannot be indexed. For performance purposes, the following nodes are automatically indexed: • Primary keys nodes. See primary keys [p 517]. • Nodes defining a foreign key constraint. See foreign key constraint [p 522]. • Nodes declared as being unique. See uniqueness constraint [p 539]. • Auto-incremented nodes. See auto-incremented values [p 553].	
`recordForm`	Defines a specific component for customizing the record form in a dataset. This component is defined using a JavaBean that extends `UIForm`[API] or implements `UserServiceRecordFormFactory`[API]. The JavaBean is specified by means of the attribute `osd:class`, for example: `<recordForm osd:class="com.wombat.MyRecordForm"/>`	No.

Example

Below is an example of a product catalog:

```
<xs:element name="Products" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
   <xs:documentation>
    <osd:label>Product Table </osd:label>
    <osd:description>List of products in Catalog </osd:description>
   </xs:documentation>
  </xs:annotation>
  <xs:complexType>
  <xs:annotation>
   <xs:appinfo>
    <osd:table>
     <primaryKeys>./productRange /productCode</primaryKeys>
     <index name="indexProductCode">/productCode</index>
    </osd:table>
   </xs:appinfo>
  </xs:annotation>
   <xs:sequence>
    <xs:element name="productRange" type="xs:string"/><!-- key -->
    <xs:element name="productCode" type="xs:string"/><!-- key -->
    <xs:element name="productLabel" type="xs:string"/>
    <xs:element name="productDescription" type="xs:string"/>
    <xs:element name="productWeight" type="xs:int"/>
    <xs:element name="productType" type="xs:string"/>
    <xs:element name="productCreationDate" type="xs:date"/>
   </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Catalogs" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
   <xs:documentation>
    <osd:label>Catalog Table</osd:label>
    <osd:description>List of catalogs</osd:description>
   </xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:annotation>
    <xs:appinfo>
     <osd:table>
      <primaryKeys>/catalogId</primaryKeys>
```

```
     </osd:table>
    </xs:appinfo>
   </xs:annotation>
   <xs:sequence>
    <xs:element name="catalogId" type="xs:string"/><!-- key -->
    <xs:element name="catalogLabel" type="xs:string"/>
    <xs:element name="catalogDescription" type="xs:string"/>
    <xs:element name="catalogType" type="xs:string"/>
    <xs:element name="catalogPublicationDate" type="xs:date"/>
   </xs:sequence>
  </xs:complexType>
</xs:element>
```

Properties related to dataset inheritance

The following properties are only valid in the context of dataset inheritance:

Element	Description	Required
`onDelete-deleteOccultingChildren`	Indique si lors de la suppression d'un enregistrement les enregistrements occultants doivent aussi tre supprims dans les jeux de donnes enfants. Valid values are: `never` or `always`.	No, default is `never`.
`mayCreateRoot`	Specifies whether root record creation is allowed. The expression must follow the syntax below. See definition modes [p 296].	No, default is `always`.
`mayCreateOverwriting`	Indique si un enregistrement peut tre surcharg dans des jeux de donnes enfants. The expression must follow the syntax below. See definition modes [p 296].	No, default is `always`.
`mayCreateOcculting`	Indique si un enregistrement occultant peut tre cr dans des jeux de donnes enfants. The expression must follow the syntax below. See definition modes [p 296].	No, default is `always`.
`mayDuplicate`	Indique si un enregistrement peut tre dupliqu. The expression must follow the syntax below.	No, default is `always`.
`mayDelete`	Indique si un enregistrement peut tre supprim. The expression must follow the syntax below.	No, default is `always`.

The `may...` expressions specify when the action is possible, though the ultimate availability of the action also depends on the user access rights. The expressions have the following syntax:

```
expression ::= always | never | <condition>*
condition ::= [root:yes | root:no]
"always": the operation is "always" possible (but user rights may restrict this).
"never": the operation is never possible.
"root:yes": the operation is possible if the record is in a root instance.
"root:no": the operation is not possible if the record is in a root instance.
```

If the record does not define any specific conditions, the default is used.

Voir aussi*Hritage de jeu de donnes [p 295]*

Using toolbars

It is possible to define the toolbars to display in the user interface using the element `defaultView/toolbars` under `xs:annotation/appinfo/osd:table`. A toolbar allows to customize the buttons and menus to display when displaying a table view, a hierarchical view, or a record form.

The table below presents the elements that can be defined under `defaultView/toolbars`.

Element	Description	Required
`tabularViewTop`	Dfinit la barre d'outils afficher en entte de la vue table par dfaut.	No.
`tabularViewRow`	Dfinit la barre d'outils afficher pour chaque ligne de la vue table par dfaut.	No.
`recordTop`	Dfinit la barre d'outils afficher en entte du formulaire d'un enregistrement.	No.
`hierarchyViewTop`	Dfinit la barre d'outils afficher en entte de la hirarchie par dfaut de la table.	No.

Voir aussi*Toolbars* *[p 571]*

Example

Below is an example of custom toolbars used by a product catalog:

```
<xs:element name="Products" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
   <xs:documentation>
    <osd:label>Product Table </osd:label>
    <osd:description>List of products in Catalog </osd:description>
   </xs:documentation>
  </xs:annotation>
  <xs:complexType>
  <xs:annotation>
   <xs:appinfo>
    <osd:table>
     <primaryKeys>./productRange /productCode</primaryKeys>
    <defaultView>
     <toolbars>
      <tabularViewTop>toolbar_name_for_tabularViewTop</tabularViewTop>
      <tabularViewRow>toolbar_name_for_tabularViewRow</tabularViewRow>
      <recordTop>toolbar_name_for_recordTop</recordTop>
      <hierarchyViewTop>toolbar_name_for_hierarchyViewTop</hierarchyViewTop>
     </toolbars>
    </defaultView>
   </osd:table>
   </xs:appinfo>
  </xs:annotation>
 ...
  </xs:complexType>
</xs:element>
```

> **Note**
>
> If a toolbar does not exist or is not available for a specific location then no toolbar will be displayed in the user interface in the corresponding location.

82.2 Foreign keys

Declaration

A reference to a table [p 517] is defined using the extended facet `osd:tableRef`.

The node holding the `osd:tableRef` declaration must be of type `xs:string`. At the instantiation, any value of the node identifies a record in the target table using its **primary key syntax** `PrimaryKey`.

`syntax`[API]. This extended facet is also interpreted as an enumeration whose values refer to the records in the target table.

Element	Description	Required
`tablePath`	XPath expression that specifies the target table.	Yes.
`container`	Reference of the dataset that contains the target table.	Only if the dataspace element is defined. Otherwise, default is the current dataset.
`branch`	Reference of the dataspace that contains the container dataset.	No, default is the current dataspace or snapshot.

`display`	Custom display for presenting the selected foreign key in the current record and the sorted list of possible keys. Two variants can be specified, either pattern-based expressions, or a JavaBean if the needs are very specific:
	• Static expressions are specified using the `display` and `pattern` elements. These static expressions can be localized using the additional attribute `xml:lang` on the `pattern` element, for example:
	`<display>`
	`<pattern>Product : ${./productCode}</pattern>`
	`<pattern xml:lang="fr-FR">Produit : ${./productCode}</pattern>`
	`<pattern xml:lang="en-US">Product: ${./productCode}</pattern>`
	`</display>`
	• A JavaBean that implements the interface `TableRefDisplay`[API]. It is specified using the attribute `osd:class`. For example:
	`<display osd:class="com.wombat.MyLabel"></display>`
	It is not possible to define both variants on the same foreign key element.
	Attention: Les droits d'accs dfinis dans les jeux de donnes associs ne sont pas appliqus lors de l'affichage des libells. Les champs habituellement cachs cause de droits d'accs seront affichs dans les libells des enregistrements.
		No, if the `display` property is not specified, the table's record rendering [p 519] is used.

`filter`	Specifies an additional constraint that filters the records of the target table. Two types of filters are available: • An XPath filter is an XPath predicate in the target table context. It is specified using the `predicate` element. For example: `<filter><predicate>type = ${../refType}</predicate></filter>` A localized validation message can be specified using the element `validationMessage`, which will be displayed to the end-user at the validation time if a record is not accepted by the filter. A specific severity level can be defined in a nested `severity` element. The default severity is 'error'. Each localized message variant is defined in a nested `message` element with its locale in an `xml:lang` attribute.	No.

Element	Description	Required
	To specify a default message for unsupported locales, define a `message` element with no `xml:lang` attribute. In the user interface, the XPath filter is applied to filter a table according to the value of a foreign key field. That is, if a foreign key field specifies an XPath filter in a data model, then it will be reused in the filter pane to restrict the set of values in the associated combo-box displayed in the filter pane. However, the predicate used by the filter pane will only take into account the non-contextual parts of the predicate. • A programmatic filter is a JavaBean that implements the interface `TableRefFilter`[API]. It is specified using the attribute `osd:class`. For example: `<filter osd:class="com.wombat.MyFilter"></filter>` Additional validation messages can be specified during the setup of the programmatic filter. In the user interface, programmatic filters are not applied to filter the set of values in the associated combo-box displayed in the filter pane. However, it is possible to specify an additional XPath predicate that will be used in the filter pane of the user interface. This XPath predicate is specified during the setup of the programmatic filter using the method `TableRefFilterContext.setFilterForSearch`[API]. **Note:** The attributes `osd:class` and the property `predicate` cannot be set simultaneously. The validation search XPath functions are forbidden on a `tableRef` filter.	
`validation`	Specifies localized validation messages for the `osd:tableRef` and error management policy. A specific severity level can be defined in a nested `severity` element. The default severity is 'error'. An error management policy can be defined in a nested `blocksCommit` element. The error management policy that blocks all operations does not apply to filters. That is, a foreign key constraint is not blocking if a referenced record exists but does not satisfy a foreign key filter. In this case, updates are not rejected, and a validation error will be reported. Each localized message variant is defined in a nested `message` element with its locale in an `xml:lang` attribute. To specify a default message for unsupported locales, define a `message` element with no `xml:lang` attribute.	No.

Attention

You can create a dataset which has a foreign key to a container that does not exist in the repository. However, the content of this dataset will not be available until the container is created. After the creation of the container, a data model refresh is required to make the dataset available. When creating a dataset that refers to a container that does not yet exist, the following limitations apply:

- Triggers defined at the dataset level are not executed.
- Default values for fields that are not contained in tables are not initialized.

- During an archive import, it is not possible to create a dataset that refers to a container that does not exist.

Example

The example below specifies a foreign key in the 'Products' table to a record of the 'Catalogs' table.

```
<xs:element name="catalog_ref" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:tableRef>
     <tablePath>/root/Catalogs</tablePath>
     <display>
       <pattern xml:lang="en-US">Catalog: ${./catalogId}</pattern>
       <pattern xml:lang="fr-FR">Catalogue : ${./catalogId}</pattern>
     </display>
      <validation>
       <severity>error</severity>
       <blocksCommit>onInsertUpdateOrDelete</blocksCommit>
       <message>A default error message</message>
       <message xml:lang="en-US">A localized error message</message>
       <message xml:lang="fr-FR">Un message d'erreur localisé</message>
     </validation>
    </osd:tableRef>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Voir aussi

Table definition [p 517]

Primary key syntax `PrimaryKey.syntax`[API]

Extraction of foreign keys (XPath predicate syntax) [p 255]

Associations [p 525]

View for advanced selection [p 565]

`SchemaNode.getFacetOnTableReference`[API]

`SchemaFacetTableRef`[API]

82.3 Associations

Overview

An association provides an abstraction over an existing relationship in the data model, and allows an easy model-driven integration of *associated objects* in the user interface and in data services.

Several types of associations are supported:

- *'By foreign key'* specifies the inverse relationship of an existing foreign key field [p 522].
- *'Over a link table'* specifies a relationship based on an intermediate link table (such tables are often called "join tables"). This link table has to define two foreign keys, one referring to the 'source' table (the table holding the association element) and another one referring to the 'target' table.
- *'By an XPath predicate'* specifies a relationship based on an XPath predicate.

For an association, it is also possible to:

- Filter associated objects by specifying an additional XPath filter.

- Configure a tabular view to define the fields that must be displayed in the associated table.
- Define how associated objects are to be rendered in forms.
- Hide/show associated objects in the data service 'select' operation. See Hiding a field in Data Services [p 564].
- Specify the minimum and maximum number of associated objects that are required.
- Add validation constraints using XPath predicates for restricting associated objects.

Voir aussi

`SchemaNode.getAssociationLink`[API]

`SchemaNode.isAssociationNode`[API]

`AssociationLink`[API]

Declaration

Associations are defined in the data model using the XML Schema element `osd:association` under `xs:annotation/appInfo`.

Restrictions:

- An association must be a simple element of type *xs:string*.
- An association can only be defined inside a table.

> **Note**
>
> The "official" cardinality constraints (`minOccurs="0" maxOccurs="0"`) are required because, from an instance of XML Schema, the corresponding node is absent. In other words, an association has no value and is considered as a "virtual" element as far as XML and XML Schema is concerned.

The table below presents the elements that can be defined under `xs:annotation/appInfo/osd:association`.

Element	Description	Required
`tableRefInverse`	Defines the properties of an association that is the inverse relationship of a *foreign key*. Element `fieldToSource` defines the foreign key that refers to the source table of the association. The element `fieldToSource` is mandatory and must specify a foreign key field that refers to the table containing the association.	Yes if the association is the inverse relationship of a *foreign key*, otherwise no.
`linkTable`	Defines the properties of an association over a link table. The element `table` specifies the link table used by the association. The element `table` is mandatory and must refer to an existing table. **Important:** In order to be used by an association, a link table must define a primary key that is composed of auto-incremented fields and/or the foreign key to the source or target table of the association. The element `fieldToSource` defines the foreign key that refers to the source table of the association. The element `fieldToSource` is mandatory and must specify a foreign key field that refers to the table containing the association. The element `fieldToTarget` defines the foreign key that refers to the target table of the association. The element `fieldToTarget` is mandatory and must specify a foreign key field.	Yes if the association is over a link table, otherwise no.
`xpathLink`	Defines the properties of an association that is based on an *XPath predicate*. The `predicate` element specifies the criteria of the association, relative to the current node. Examples: `/root/Products[catalog_ref =${../catalogId}]` or `//Products[catalog_ref = ${../catalogId}]` or `../Products[catalog_ref =${../catalogId}]`. The path to the predicate, for example `../Products`, specifies the target table of the association. This part of the path is resolved with respect to the current table. It is not possible to refer to a table using a relative path if the association targets a table in another dataset. If the association depends on fields of the source table, the XPath expression predicate must include references to the elements on which it depends using the notation `${<relative-path>}` where `relative-path` is a path that identifies the element relative to the association node. See EBX XPath supported syntax [p 249]. **Note**	Yes if the association is based on an *XPath predicate*, otherwise no.

Element	Description	Required
	The validation search XPath functions are forbidden on an XPath link.	
`filter`	Defines an XPath predicate to filter associated objects using the `predicate` element. For example: `<filter><predicate>type = ${../refType}</predicate></filter>` It is only possible to use fields from the source and the target tables when defining an XPath filter. That is, if it is an association over a link table, it is not possible to use fields of the link table in the XPath filter. **Error message on creation**: in the user interface, the record creation is blocked when a user submits a new associated record that does not comply with the filter. The error message can be customized using the element `checkOnAssociatedRecordCreation/message`. Each localized message variant is defined in a nested message element with its locale in an `xml:lang` attribute. To specify a default message for unsupported locales, define a message element with no `xml:lang` attribute. See Examples [p 534] for more information on this property. **Note** The validation search XPath functions are forbidden for association filter.	No.
`xpathFilter`	**Note:** Deprecated. This property has been replaced by the property `filter`. Defines an XPath predicate to filter associated objects.	No.
`recordForm`	Defines a specific component for customizing the form of an associated record. This component is defined using a JavaBean that implements `UserServiceAssociationRecordFormFactory`[API]. The JavaBean is specified by means of the attribute `osd:class`, for example: `<recordForm osd:class="com.wombat.MyRecordFormFactory"/>`	No.

It is possible to refer to another dataset. For that, the following properties must be defined either under the element `tableRefInverse`, `linkTable` or `xpathLink` depending on the type of the association:

Element	Description	Required
`schemaLocation`	Defines the data model containing the fields used by the association. The data model is defined using a specific URN that allows referring to embedded data models and data models packaged in modules. See `SchemaLocation`[API] for more information about specific URNs supported by EBX.	Yes.
`dataSet`	Defines the dataset used by the association. This dataset must use the data model specified by the element `schemaLocation`.	Yes.
`dataSpace`	Defines the dataspace containing the dataset used by the association.	No.

Important: When creating a dataset, you can create a dataset that defines an association to a container that does not yet exist in the repository. However, the content of this dataset will not be available immediately upon creation. After the absent container is created, a data model refresh is required in order to make the dataset available. When creating a dataset that refers to a container that does not yet exist, the following limitations apply:

- Triggers defined at the dataset level are not executed.
- Default values on fields outside tables are not initialized.
- During an archive import, it is not possible to create a dataset that refers to a container that does not exist.

User interface integration

It is possible to define how associated objects are to be rendered in forms, using the element `osd:defaultView/displayMode` under `xs:annotation/appinfo`.

Possible values are:

- `inline`, specifies that associated records are to be rendered in the form at the same position of the association in the data model.
- `tab`, specifies that associated records are to be rendered in a specific tab.
- `link`, specifies that associated records are to be rendered in a modal window.

By default, associated records are rendered `inline` if this property is not defined.

The following example specifies that associated objects are to be rendered `inline` in the form:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
  <osd:association>
   <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
   </tableRefInverse>
  </osd:association>
  <osd:defaultView>
   <displayMode>inline</displayMode>
  </osd:defaultView>
   </xs:appinfo>
   </xs:annotation>
```

```
</xs:element>
```

The following example specifies that associated objects are to be rendered in a specific tab:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
  <osd:association>
   <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
   </tableRefInverse>
  </osd:association>
  <osd:defaultView>
   <displayMode>tab</displayMode>
  </osd:defaultView>
   </xs:appinfo>
   </xs:annotation>
</xs:element>
```

Using toolbars

It is possible to define the toolbars to display in the user interface using the element `osd:defaultView/toolbars` under `xs:annotation/appinfo`. A toolbar allows to customize the buttons and menus to display when displaying the tabular view of an association.

The table below presents the elements that can be defined under `osd:defaultView/toolbars`.

Element	Description	Required
`tabularViewTop`	Dfinit la barre d'outils afficher en entte de la vue table par dfaut de cette association.	No.
`tabularViewRow`	Dfinit la barre d'outils afficher pour chaque ligne de la vue table par dfaut de cette association.	No.

The following example shows how to use toolbars from the previous association between a catalog and its products:

```
<xs:element name="Products" minOccurs="0" maxOccurs="0" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:association>
    <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
    </tableRefInverse>
   </osd:association>
   <osd:defaultView>
    <toolbars>
     <tabularViewTop>toolbar_name_for_tabularViewTop</tabularViewTop>
     <tabularViewRow>toolbar_name_for_tabularViewRow</tabularViewRow>
    </toolbars>
   </osd:defaultView>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

> **Note**
>
> It is only possible to use the toolbars defined in the data model containing the target table of the association. That is, if the target table of the association is defined in another data model, then it is only possible to reference a toolbar defined in this data model and not in the one holding the association.

Voir aussi*Toolbars* *[p 571]*

Customized view of associated objects

A specific tabular view can be specified to define the fields that must be displayed in the target table. If a tabular view is not defined, all columns that a user is allowed to view, according to the granted access rights, are displayed. A tabular view is defined using the element `osd:defaultView/tabularView` under `xs:annotation/appinfo`.

The table below shows the elements that can be defined under `osd:defaultView/tabularView`.

Element	Description	Required
`column`	Define a field of the target table to display. The specified path must be absolute from the target table and must refer to an existing field. Several `column` elements can be defined to specify the fields that are to be displayed.	No.
`sort`	Define a field that can be used to sort associated objects. Several`sort` elements can be defined to specify the fields that can be used to sort associated objects. The element `nodePath` defines the path of the field that can be used to sort associated objects. The element `isAscending` specifies whether the sort order is ascending (true) or descending (false).	No.

The following example shows how to define a tabular view from the previous association between a catalog and its products:

```
<xs:element name="Products" minOccurs="0" maxOccurs="0" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:association>
    <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
    </tableRefInverse>
   </osd:association>
   <osd:defaultView>
    <tabularView>
     <column>/productRange</column>
     <column>/productCode</column>
     <column>/productLabel</column>
     <column>/productDescription</column>
     <sort>
        <nodePath>/productLabel</nodePath>
        <isAscending>true</isAscending>
     </sort>
    </tabularView>
   </osd:defaultView>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Actions in the user interface

In the user interface, it is possible to perform the following actions:

- **Create**: it allows directly creating an object in the target table of the association. When a new object is created, it is automatically associated with the current record.
- **Duplicate**: allows to duplicate an object in the target table of the association. When a new object is created, it is automatically associated with the current record.

- **Associate**: associates an existing object with the current record. In the case of an association over a link table, a record in the link table is automatically created to materialize the link between the current record and the existing object.
- **Move**: associates the selected objects to a different record than the current one. In the case of an association over a link table, the previous link record is automatically deleted and a new record in the link table is automatically created to materialize the link between the selected objects and their new parent record.
- **Delete**: deletes selected associated objects in the target table of the association.
- **Detach**: breaks the semantic link between the current record and the selected associated objects. In the case of an association over a link table, the records in the link table are automatically deleted, to break the links between the current record and associated objects.

> **Note**
>
> The actions *associate* and *detach* are not available when the association is defined using an **XPath predicate** (element *xpathLink*).

Customized view for actions

A published view, tabular or hierarchical, can be specified to define how objects should be displayed when performing an action through the user interface. A published view is defined using the element `osd:defaultView/associationViews` under `xs:annotation/appinfo`.

The table below shows the elements that can be defined under `osd:defaultView/associationViews`.

Element	Description	Required
`viewForAssociateAction`	Define a published view to be used when displaying the objects in the target table to be associated with the current record. The specified view must be published and created upon the target table of the association.	No.
`viewForMoveAction`	Define a published view to be used when moving an associated object to another record of the current table. The specified view must be published and created upon the current table.	No.

The following example shows how to define views from the previous association between a catalog and its products:

```
<xs:element name="Products" minOccurs="0" maxOccurs="0" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:association>
    <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
    </tableRefInverse>
   </osd:association>
   <osd:defaultView>
    <associationViews>
     <viewForAssociateAction>view_name_for_catalogs</viewForAssociateAction>
     <viewForMoveAction>view_name_for_products</viewForMoveAction>
    </associationViews>
   </osd:defaultView>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Validation

Some controls can be defined on associations, in order to restrict associated objects. These controls are defined under the element `osd:association`.

The table below presents the controls that can be defined under xs:annotation/appInfo/osd:association.

Element	Description	Required
`minOccurs`	Specifies the minimum number of associated objects that are required for this association. This minimum number is defined using the element `value` and must be a positive integer.	No, by default the minimum is not restricted.
`maxOccurs`	Specifies the maximum number of associated objects that are allowed for this association. This maximum number is defined by the element `value` and must be either a positive integer or the raw string `unbounded` which indicates that this maximum is not restricted. The maximum number of associated objects must be greater than the minimum number of associated objects.	No, by default the maximum is not restricted.
`constraint`	Defines an XPath predicate for restricting associated records. It is only possible to use fields from the source and the target table when defining an XPath predicate. That is, if it is an association over a link table, it is not possible to use fields of the link table in the XPath predicate. In associated datasets, a validation message of the specified severity is added and displayed to the end-user at the validation time when an associated record does not comply with the specified constraint.	No.
`validation`	A validation message can be defined under the elements `minOccurs`, `maxOccurs` and `constraint`, using the element `validation`. The severity of the validation message is specified using the element `severity`. Possible severities are: `error`, `warning` and `info`. If the severity is not specified then, by default, the severity `error` is used. A localized validation message can be specified using the element `message`, which will be displayed to the end-user at the validation time if an association does not comply with this constraint. Each localized message variant is defined in a nested message element with its locale in an `xml:lang` attribute. To specify a default message for unsupported locales, define a message element with no `xml:lang` attribute.	No.

Data services integration

It is possible to define whether associated objects must be hidden in the Data service `select` operation. For this, the property `osd:defaultView/hiddenInDataServices` under `xs:annotation/xs:appinfo` can be set on the association. Setting the property to 'true' will hide associated objects in the Data

service `select` operation. If this property is not defined then, by default, associated objects will be shown in the Data service `select` operation.

Voir aussi

Hiding a field in Data Services *[p 564]*

Association field *[p 640]*

Examples

For example, the product catalog data model defined previously [p 520] specifies that a product belongs to a catalog (explicitly defined by a foreign key in the 'Products' table). The reverse relationship (that a catalog has certain products) is not easily represented in XML Schema, unless the 'Catalogs' table includes the following association that is the inverse of a foreign key:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
   <osd:association>
   <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
   </tableRefInverse>
   </osd:association>
   </xs:appinfo>
   </xs:annotation>
</xs:element>
```

For an association over a link table, we can consider the previous example and bring some updates. For instance, the foreign key in the 'Products' table is deleted and the relation between a product and a catalog is redefined by a link table (named 'Catalogs_Products') that has a primary key composed of two foreign keys: one that refers to the 'Products' table (named 'productRef') and another to the 'Catalogs' table (named 'catalogRef'). The following example shows how to define an association over a link table from this new relationship:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
  <osd:association>
   <linkTable>
     <table>/root/Catalogs_Products</table>
     <fieldToSource>./catalogRef</fieldToSource>
     <fieldToTarget>./productRef</fieldToTarget>
   </linkTable>
  </osd:association>
   </xs:appinfo>
   </xs:annotation>
</xs:element>
```

The following example shows an association that refers to a foreign key in another dataset. In this example, the 'Products' and 'Catalogs' tables are not in the same dataset:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
   <osd:association>
   <tableRefInverse>
    <schemaLocation>urn:ebx:module:aModuleName:/WEB-INF/ebx/schema/products.xsd</schemaLocation>
    <dataSet>Products</dataSet>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
   </tableRefInverse>
   </osd:association>
   </xs:appinfo>
   </xs:annotation>
</xs:element>
```

The following example defines an XPath filter to associate only products of the 'Technology' type:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
   <xs:annotation>
   <xs:appinfo>
```

```
   <osd:association>
   <tableRefInverse>
    <fieldToSource>/root/Products/catalog_ref</fieldToSource>
   </tableRefInverse>
   <filter>
    <predicate>./productType = 'Technology'</predicate>
    <checkOnAssociatedRecordCreation>
     <message>A default message</message>
                  <message xml:lang="en-US">A localized message</message>
                  <message xml:lang="fr-FR">Un message localisé</message>
    </checkOnAssociatedRecordCreation>
   </filter>
   </osd:association>
   </xs:appinfo>
   </xs:annotation>
</xs:element>
```

The following example specifies the minimum number of products that are required for a catalog:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:association>
    <tableRefInverse>
      <fieldToSource>/root/Products/catalog_ref</fieldToSource>
    </tableRefInverse>
     <minOccurs>
     <value>1</value>
     <validation>
      <severity>warning</severity>
      <message xml:lang="en-US">One product should at least be associated to this catalog.</message>
      <message xml:lang="fr-FR">Un produit doit au moins être associé à ce catalogue.</message>
     </validation>
    </minOccurs>
   </osd:association>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

The following example specifies that a catalog must contain at most ten products:

```
<xs:element name="products" minOccurs="0" maxOccurs="0" type="xs:string">
 <xs:annotation>
  <xs:appinfo>
   <osd:association>
    <tableRefInverse>
     <fieldToSource>/root/Products/catalog_ref</fieldToSource>
    </tableRefInverse>
    <maxOccurs>
     <value>10</value>
     <validation>
      <severity>warning</severity>
      <message xml:lang="en-US">Too much products for this catalog.</message>
      <message xml:lang="fr-FR">Ce catalogue a trop de produits.</message>
     </validation>
    </maxOccurs>
   </osd:association>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

CHAPITRE 83

Constraints, triggers and functions

Facets allow you to define data constraints in your data models. TIBCO EBX supports XML Schema facets and provides extended and programmatic facets for advanced data controls.

Ce chapitre contient les sections suivantes :

1. XML Schema supported facets
2. Extended facets
3. Programmatic facets
4. Control policy
5. Triggers and functions

83.1 XML Schema supported facets

The tables below show the facets that are supported by different data types.

Key:

- **X** - Supported
- **1** - The `whiteSpace` facet can be defined, but is not interpreted by EBX
- **2** - In XML Schema, boundary facets are not allowed on the type `string`. Nevertheless, EBX allows such facets as extensions.
- **3** - The `osd:resource` type only supports the facet `FacetOResource`, which is required. See Extended Facets [p 540].

- **4 -** `osd:dataspaceKey`, `osd:datasetName` and `osd:color` types do not support facets. Only Programmatic constraints [p 544] are supported on these types.

	length	minLength	max Length	pattern	enumeration	white Space
xs:string	X	X	X	X	X	1
xs:boolean				X		1
xs:decimal				X	X	1
xs:dateTime				X	X	1
xs:time				X	X	1
xs:date				X	X	1
xs:anyURI	X	X	X	X	X	1
xs:Name	X	X	X	X	X	1
xs:integer				X	X	1
osd:resource [p 506][3]						1
osd:dataspaceKey [p 508][4]						1
osd:datasetName [p 510][4]						1
osd:color [p 508][4]						1

	fraction Digits	total Digits	max Inclusive	max Exclusive	min Inclusive	min Exclusive
xs:string			2	2	2	2
xs:boolean						
xs:decimal	X	X	X	X	X	X
xs:dateTime			X	X	X	X
xs:time			X	X	X	X
xs:date			X	X	X	X

	fraction Digits	total Digits	max Inclusive	max Exclusive	min Inclusive	min Exclusive
xs:anyURI						
xs:Name			**2**	**2**	**2**	**2**
xs:integer	**X**	**X**	**X**	**X**	**X**	**X**
osd:resource [p 506][3]						
osd:dataspaceKey [p 508][4]						
osd:datasetName [p 510][4]						
osd:color [p 508][4]						

Example:

```
<xs:element name="loanRate">
 <xs:simpleType>
  <xs:restriction base="xs:decimal">
   <xs:minInclusive value="4.5" />
   <xs:maxExclusive value="17.5" />
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

Uniqueness constraint

It is possible to define a uniqueness constraint, using the standard XML Schema element xs:unique. This constraint indicates that a value or a set of values has to be unique inside a table.

Example:

In the example below, a uniqueness constraint is defined on the 'publisher' table, for the target field 'name'. This means that no two records in the 'publisher' table can have the same name.

```
<xs:element name="publisher">
 ...
 <xs:complexType>
  <xs:sequence>
   ...
   <xs:element name="name" type="xs:string" />
   ...
  </xs:sequence>
 </xs:complexType>
 <xs:unique name="uniqueName">
  <xs:annotation>
   <xs:appinfo>
    <osd:validation>
     <severity>error</severity>
     <message>Name must be unique in table.</message>
     <message xml:lang="en-US">Name must be unique in table.</message>
     <message xml:lang="fr-FR">Le nom doit être unique dans la table.</message>
    </osd:validation>
   </xs:appinfo>
  </xs:annotation>
  <xs:selector xpath="." />
  <xs:field xpath="name" />
 </xs:unique>
</xs:element>
```

A uniqueness constraint has to be defined within a table and has the following properties:

Property	Description	Mandatory
`name` attribute	Identifies the constraint in the data model.	Yes
`xs:selector` element	Indicates the table to which the uniqueness constraint applies using a restricted XPath expression ('..' is forbidden). It can also indicate an element within the table (without changing the meaning of the constraint).	Yes
`xs:field` element	Indicates the field in the context whose values must be unique, using a restricted XPath expression. It is possible to indicate that a set of values must be unique by defining multiple `xs:field` elements.	Yes

> **Note**
>
> Undefined values (null values) are ignored on uniqueness constraints applied to single fields. On multiple fields, undefined values are taken into account. That is, sets of values are considered as being duplicated if they have the same defined and undefined values.

Additional localized validation messages can be defined using the element `osd:validation` under the elements `annotation/appinfo`. If no custom validation messages are defined, a built-in validation message will be used.

Limitations:

1. The target of the `xs:field` element must be in a table.
2. The uniqueness constraint does not apply to fields inside an aggregated list.
3. The uniqueness constraint does not apply to computed fields.

Voir aussi*Uniqueness constraint in the Java API* `UniquenessConstraint`[API]

83.2 Extended facets

EBX provides additional constraints that are not specified in XML Schema, but that are useful for managing master data.

In order to guarantee XML Schema conformance, these extended facets are defined under the element `annotation/appinfo/otherFacets`.

Foreign keys

EBX allows to create a reference to an existing table by means of a specific facet. See Foreign keys [p 522] for more information.

Dynamic constraints

Dynamic constraint facets retain the semantics of XML Schema, but the value attribute is replaced with a path attribute that allows fetching the value from another element. The available dynamic constraints are:

- `length`

- `minLength`
- `maxLength`
- `maxInclusive`
- `maxExclusive`
- `minInclusive`
- `minExclusive`

Using these facets, the data model can be modified dynamically.

Example:

```
<xs:element name="amount">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:minInclusive path="/domain/Loan/Pricing/AmountMini/amount" />
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
   ...
</xs:element>
```

In this example, the boundary of the facet `minInclusive` is not statically defined. The value of the boundary comes from the node */domain/Loan/Pricing/AmountMini/amount*.

Restrictions:

- Target field cannot be an aggregated list. That is, it cannot define `maxOccurs` = 1.
- Data type of the target field must be compatible with the facet. That is, it must be:
  - of type `integer` for facets `length`, `minLength` and `maxLength`.
  - compatible with the data type of the field holding the facet for facets `maxInclusive`, `maxExclusive`, `minInclusive` and `minExclusive`.
- Target field cannot be in a table if the field holding the facet is not in a table.
- Target field must be in the same table or outside a table if the field holding the facet is in a table.
- If the target field is under one or more aggregated lists, the field holding the facet must also be under these aggregated lists. That is: the field holding the facet must be in the same list occurrence as the target field, or in a parent occurrence, so that the target field refers to a single value, from an XPath perspective.

FacetOResource constraint

This facet must be defined for every definition using the type `osd:resource`, to specify the subset of available packaged resource files as an enumeration. For more information on this type, see osd:resource type [p 508]. It has the following attributes:

`moduleName`	Indicates, using an alias, the EBX module that contains the resource. If the resource is contained in the current module, the alias must be preceded by "wbp". Otherwise, the alias must be one of the values defined in the element `<dependencies>` in the file `module.xml`.
`resourceType`	Dfinit le type de ressource parmi les valeurs suivantes : "Image", "JavaScript", "Feuille de style", "HTML".
`relativePath`	Prcise dans quel rpertoire sont contenues les ressources. Le rpertoire utilis doit tre situ sous celui correspondant au type de ressource. Par exemple, pour une ressource de type "Image", le rpertoire www/common/images/, situ au mme niveau que le rpertoire WEB-INF/ du module cible, sera utilis et le chemin relatif devra donc tre dfini partir de celui-ci. De plus, si une ressource est situe dans un rpertoire internationalis (www/fr/ par exemple), elle sera utilise uniquement si une ressource du mme nom est dfinie dans le rpertoire www/common/.

This facet has the same behavior as an enumeration facet: the values are collected by recursively listing all the files in the local path in the specified resource type directory in the specified module.

Example:

```
<xs:element name="promotion" type="osd:resource">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:FacetOResource osd:moduleName="wbp"
     osd:resourceType="ext-images" osd:relativePath="promotion/" />
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
</ xs:element>
```

For an overview of the standard directory structure of an EBX module (Java EE web application), see Module structure [p 483].

excludeValue constraint

This facet verifies that a value is not the same as the specified excluded value.

In this example, the empty string is excluded from the allowed values.

Example:

```
<xs:element name="roleName">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:excludeValue value="">
     <osd:validation>
```

```
      <severity>error</severity>
       <message>Please select address role(s).</message>
     </osd:validation>
    </osd:excludeValue>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
 <xs:simpleType type="xs:string" />
</xs:element>
```

excludeSegment constraint

This facet verifies that a value is not included in a range of values. Boundaries are excluded.

Example:

In this example, values between 20000 and 20999 are not allowed.

```
<xs:element name="zipCode">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:excludeSegment minValue="20000" maxValue="20999">
     <osd:validation>
      <severity>error</severity>
       <message>Postal code not valid.</message>
     </osd:validation>
    </osd:excludeSegment>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
 <xs:simpleType type="xs:string" />
</xs:element>
```

Enumeration facet defined using another node

By default, an enumeration facet is described statically in XML Schema.

The content of an enumeration facet can also be provided dynamically by a list of simple elements in the data model.

Example:

In this example, the content of an enumeration facet is sourced from the node `CountryList`.

```
<xs:annotation>
 <xs:appinfo>
  <osd:otherFacets>
   <osd:enumeration osd:path="../CountryList" />
  </osd:otherFacets>
 </xs:appinfo>
</xs:annotation>
```

The referred node `CountryList`:

- Must be an aggregated list, that is, `maxOccurs` > 1.
- Must be a list of elements of the same type as the node with the enumeration facet.
- Must be a node outside a table if the node with the enumeration facet is not inside a table.
- Must be a node outside a table or in the same table as the node with the enumeration facet if the node with this enumeration is inside a table.
- If the target field is under one or more aggregated lists, the field holding the facet must also be under these aggregated lists. That is: the field holding the facet must be in the same list occurrence as the target field, or in a parent occurrence, so that the target field refers to a single value, from an XPath perspective.

Example:

```
<xs:element name="FacetEnumBasedOnList">
 <xs:complexType>
```

```
  <xs:sequence>
   <xs:element name="CountryList" maxOccurs="unbounded">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="DE" osd:label="Germany" />
      <xs:enumeration value="AT" osd:label="Austria" />
      <xs:enumeration value="BE" osd:label="Belgium" />
      <xs:enumeration value="JP" osd:label="Japan" />
      <xs:enumeration value="KR" osd:label="Korea" />
      <xs:enumeration value="CN" osd:label="China" />
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="CountryChoice" type="xs:string">
    <xs:annotation>
     <xs:appinfo>
      <osd:otherFacets>
       <osd:enumeration osd:path="../CountryList" />
      </osd:otherFacets>
     </xs:appinfo>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

83.3 Programmatic facets

A programmatic constraint can be added to any XML element declaration for a simple type.

In order to guarantee XML Schema conformance, programmatic constraints are specified under the element `annotation/appinfo/otherFacets`.

Programmatic constraints

A programmatic constraint is defined by a Java class that implements the interface `Constraint`[API].

As additional parameters can be defined, the implemented Java class must conform to the JavaBean protocol.

Example:

In the example below, the Java class must define the methods: `getParam1()`, `setParam1(String)`, `getParamX()`, `setParamX(String)`, etc.

```
<xs:element name="amount">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:constraint class="com.foo.CheckAmount">
     <param1>...</param1>
     <param...n>...</param...n>
    </osd:constraint>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

Voir aussi*JavaBean specifications* `Package com.orchestranetworks.schema.JavaBeans`[API]

Programmatic enumeration constraints

An enumeration constraint adds an ordered list of values to a basic programmatic constraint. This facet allows selecting a value from a list. It is defined by a Java class that implements the interface `ConstraintEnumeration`[API].

Example:

```
<xs:element name="amount">
 <xs:annotation>
```

```
  <xs:appinfo>
   <osd:otherFacets>
    <osd:constraintEnumeration class="com.foo.CheckAmountInEnumeration">
     <param1>...</param1>
     <param...n>...</param...n>
    </osd:constraintEnumeration>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

Constraint on 'null' values

In some cases, a value is only mandatory if some conditions are satisfied, for example, if another field has a given value. In this case, the standard XML Schema attribute `minOccurs` is insufficient because it is static.

In order to check if a value is mandatory according to its context, the following requirements must be satisfied:

1. A programmatic constraint must be defined by a Java class (see above).
2. This class must implement the interface `ConstraintOnNull`[API].
3. The XML Schema cardinality attributes must specify that the element is optional (`minOccurs="0"` and `maxOccurs="1"`).

> **Note**
>
> By default, constraints on 'null' values are not checked upon user input. In order to enable a check at the input, the 'checkNullInput' property [p 549] must be set. Also, if the element is terminal, the dataset must also be activated.

Example:

```
<xs:element name="amount" minOccurs="0" maxOccurs="1">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:constraint class="com.foo.CheckIfNull">
     <param1>...</param1>
     <param...n>...</param...n>
    </osd:constraint>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

Constraints on table

A constraint on table is defined by a Java class that implements the interface `ConstraintOnTable`[API]. It can only be defined on table nodes.

As additional parameters can be defined. the implemented Java class must conform to the JavaBean protocol.

Example:

In the example below, the Java class must define the methods: `getParam1()`, `setParam1(String)`, `getParamX()`, `setParamX(String)`, etc.

```
<xs:element name="myTable" type="MyTableType" minOccurs="0" maxOccurs="unbounded">
 <xs:annotation>
  <xs:appinfo>
   <osd:table>
    <primaryKeys>/key</primaryKeys>
   </osd:table>
```

```
   <osd:otherFacets>
    <osd:constraint class="com.foo.checkTable">
     <param1>...</param1>
     <param...n>...</param...n>
    </osd:constraint>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Attention

For performance reasons, constraints on tables are only checked when getting the validation report of a dataset or table. This means that these constraints are not checked when updates, such as record insertions, deletions or modifications, occur on tables. However, the internal incremental validation framework will optimize the validation cost of these constraints if dependencies are defined. For more information, see Validation [p 324].

Voir aussi*JavaBeans Specifications* `Package com.orchestranetworks.schema.JavaBeans`[API]

83.4 Control policy

Blocking and non-blocking constraints

When an update in the repository is performed, and this update adds a validation error according to a given constraint, it is possible to specify whether the new error blocks the update (and cancels the transaction) or if it is considered as non-blocking (so that the update can be committed and the error

can be corrected later). The element `blocksCommit` within the element `osd:validation` allows this specification, with the following supported values:

`onInsertUpdateOrDelete`	Specifies that the constraint must always remain valid after an operation (dataset update, dataset deletion, record creation, update or deletion). In this case, any operation that would violate the constraint is rejected and the values remain unchanged. This is the default and mandatory policy for primary key constraints, data type conversion constraints (an integer or a date must be well-written) and also structural constraints in mapped tables. This is also the default policy for foreign key constraints that are automatically set in blocking mode (because of a table in relational mode involved in the relationship) and that do not define a control policy.
`onUserSubmit-checkModifiedValues`	Specifies that the constraint must remain valid whenever a user modifies the associated value and submits a form. In this case, any form input that would violate the constraint is rejected and the values remain unchanged. This is the default policy for all blocking constraints mentioned in the previous case. For example, a foreign key constraint is by default not blocking (a record referred to by other records can be deleted, etc.), except in the context of a form submit.
`never`	Specifies that the constraint must never block operations. In this case, any operation that would violate the constraint is allowed. In the context of the user interface, this constraint does not block the form submission if the user sets a value that violates this constraint.

On foreign key constraints, the control policy that blocks all operations does not apply to filtered records. That is, a foreign key constraint is not blocking if a referenced record exists but does not satisfy a foreign key filter. In this case, updates are not rejected and a validation error occurs.

It is not possible to specify a control policy on structural constraints that are defined on relational data models or in mapped tables. That is, this property is not available for fixed length, maximum length, maximum number of digits, and decimal place constraints due to the validation policy of the underlying RDBMS blocking constraints.

This property does not apply to archive imports and when merging dataspaces. That is, all blocking constraints, except structural constraints, are always disabled when importing archives and merging dataspaces.

Voir aussi

Facet validation message with severity *[p 558]*

Foreign keys *[p 522]*

Mode relationnel [p 267]

XML Schema facet

The control policy is described by the element `osd:validation` in `annotation/appinfo` under the definition of the facet.

Example:

```
<xs:element name="zipCode">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:minInclusive value="1000">
    <xs:annotation>
     <xs:appinfo>
      <osd:validation>
       <blocksCommit>onInsertUpdateOrDelete</blocksCommit>
      </osd:validation>
     </xs:appinfo>
    </xs:annotation>
   </xs:minInclusive>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

XML Schema enumeration facet

The control policy is described by the element `osd:enumerationValidation` in `annotation/appinfo` under the definition of the field.

Example:

```
<xs:element name="Gender">
 <xs:annotation>
  <xs:appinfo>
   <osd:enumerationValidation>
    <blocksCommit>onInsertUpdateOrDelete</blocksCommit>
   </osd:enumerationValidation>
  </xs:appinfo>
 </xs:annotation>
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="0" osd:label="male" />
   <xs:enumeration value="1" osd:label="female" />
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

EBX facet

The control policy is described by the element `osd:validation` under the definition of the facet (which is defined in `annotation/appinfo/otherFacets`).

The control policy with values `onInsertUpdateOrDelete` and `onUserSubmit-checkModifiedValues` is only available on `osd:excludeSegment`, `osd:excludeValue` and `osd:tableRef` EBX facets.

The control policy with the value `never` can be defined on all EBX facets. On programmatic constraints, the control policy with the value `never` can only be set directly during the setup of the corresponding constraint. See `ConstraintContext.setBlocksCommitToNever`[API] and `ConstraintContextOnTable.setBlocksCommitToNever`[API] in the Java API for more information.

Example:

```
<xs:element name="price" type="xs:decimal">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:minInclusive path="../priceMin">
     <osd:validation>
      <blocksCommit>onInsertUpdateOrDelete</blocksCommit>
     </osd:validation>
    </osd:minInclusive>
```

```
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Check 'null' input

According to the EBX default validation policy, in order to allow temporarily incomplete input, a mandatory element is not checked for completion upon user input. Rather, it is verified at the dataset validation only. If completion must be checked immediately upon user input, the element must additionally specify the attribute `osd:checkNullInput="true"`.

> **Note**
>
> A value is mandatory if the data model specifies a mandatory element, either statically, using `minOccurs="1"`, or dynamically, using a constraint on 'null'. For terminal elements, mandatory values are only checked for an activated dataset. For non-terminal elements, the dataset does not need to be activated.

Example:

```
<xs:element name="amount" osd:checkNullInput="true" minOccurs="1">
 ...
</xs:element>
```

Voir aussi

Constraint on 'null' *[p 545]*

Whitespace management *[p 549]*

Empty string management *[p 551]*

EBX whitespace management for data types

According to XML Schema (see https://www.w3.org/TR/xmlschema-2/#rf-whiteSpace), whitespace handling must follow one of the procedures *preserve*, *replace* or *collapse*:

preserve	No normalization is performed, the value is unchanged.
replace	All occurrences of `#x9` (tab), `#xA` (line feed) and `#xD` (carriage return) are replaced with `#x20` (space).
collapse	After the processing according to the replace procedure, contiguous sequences of `#x20` are then collapsed to a single `#x20`, and any leading or trailing `#x20`s are removed.

General whitespace handling

EBX complies with the XML Schema recommendation:

- For fields of type `xs:string`, whether a primary key element or not, whitespaces are always preserved and an empty string is never converted to `null`.

- For other fields (non-`xs:string` type), whitespaces are always collapsed and empty strings are converted to `null`.

Attention

Exceptions:

- For fields of type `osd:html` or `osd:password`, whitespaces are always preserved and empty strings are converted to `null`.
- For fields of type `xs:string` that define the property `osd:checkNullInput="true"`, an empty string is interpreted as `null` at user input by EBX.

Whitespace handling upon user input

The rules described in the previous section are applied in the user interface, but leading and trailing whitespaces are removed upon user input. That is, in the user interface, whitespaces are by default always trimmed upon user input. Other input methods (Import XML/CSV, Data services, API updates) are not trimmed from the user interface.

Attention

Exceptions:

- For fields of type `osd:password`, whitespaces are not trimmed upon user input.
- For foreign key fields, whitespaces are not trimmed upon user input.

It is possible to indicate in a data model that whitespaces should not be trimmed upon user input. The attribute `osd:trim="disable"` can be set on the fields that allow leading and trailing whitespaces upon user input.

Example:

```
<xs:element name="field" osd:trim="disable" type="xs:string">
 ...
</xs:element>
```

Whitespace handling for primary keys of type string

For primary key columns of type `xs:string`, a default EBX constraint is defined. This constraint forbids empty strings and non-collapsed whitespace values when creating a record. That is, any record creation that would violate this constraint is rejected.

However, if the primary key node specifies its own `xs:pattern` facet, this facet overrides the default EBX constraint. For example, the specific pattern `".*"` would accept any string, although this is not recommended.

The default constraint allows handling certain ambiguities. For example, it would be difficult for a user to distinguish between the following strings: "12 34" and "12 34". For generic values, this would not create conflicts, however, errors would occur for primary keys.

Voir aussi*Tables and relationships* *[p 517]*

Empty string management

Default conversion

For nodes of type `xs:string`, no distinction is made at user input between an empty string and a `null` value. That is, an empty string value is automatically converted to `null` at user input.

Distinction between empty strings and 'null' value

There are certain cases where the distinction is made between an empty string and the `null` value, such as when:

- A primary key defines a pattern that allows empty strings.
- An element defines a foreign key constraint and the target table has a single primary key defining a pattern that allows empty strings.
- An element defines a static enumeration that contains an empty string.
- An element defines a dynamic enumeration to another element with one of the aforementioned cases.

If the distinction is made between an empty string and a `null` value, this implies the following behaviors:

- An empty string will not be converted to `null` at user input,
- Input fields for nodes of type `xs:string` will display an additional button for setting the value of the node to `null`,
- At validation time, an empty string will be considered to be a compliant value with regard to the `minOccurs="1"` property.

Attention

In relational mode, the Oracle database does not support the distinction between empty strings and `null` values, and these specific cases are not supported.

Voir aussi*Mode relationnel* *[p 267]*

83.5 Triggers and functions

Computed values

By default, data is read and persisted in the XML repository. Nevertheless, data may be the result of a computation and/or external database access, for example, an RDBMS or a central system.

EBX allows taking into account other data in the current dataset context.

This is made possible by defining *functions*.

A function is specified in the data model using the `osd:function` element (see example below).

- The value of the *class* attribute must be the qualified name of a Java class that implements the Java interface `ValueFunction`[API].

- Additional parameters may be specified at the data model level, in which case the JavaBean convention is applied.

Example:

```
<xs:element name="computedValue">
 <xs:annotation>
  <xs:appinfo>
   <osd:function class="com.foo.ComputeValue">
    <param1>...</param1>
    <param...n>...</param...n>
   </osd:function>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

In some cases, it can be useful to disable the validation of computed values if the execution of a function is time-consuming. Indeed, if the function is attached to a table with N records, then it will be called N times when validating this table. The property `osd:disableValidation= "true"` specified in the data model allows to disable the validation of a computed value (see example below).

Example:

```
<xs:element name="computedValue" osd:disableValidation="true">
 <xs:annotation>
  <xs:appinfo>
   <osd:function class="com.foo.ComputeValue">
    <param1>...</param1>
    <param...n>...</param...n>
   </osd:function>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

Triggers

Datasets or table records can be associated with methods that are automatically executed when some operations are performed, such as creations, updates, or deletions.

In the data model, these triggers must be declared under the `annotation/appinfo` element using the `osd:trigger` element.

For dataset triggers, a Java class that extends the abstract class `InstanceTrigger`[API] must be declared inside the element `osd:trigger`.

In the case of dataset triggers, it is advised to define `annotation/appinfo/osd:trigger` tags just under the `root` element of the data model.

Example:

```
<xs:element name="root" osd:access="--">
   ...
   <xs:annotation>
  <xs:appinfo>
   <osd:trigger class="com.foo.MyInstanceTrigger">
    <param1>...</param1>
    <param...n>...</param...n>
   </osd:trigger>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:element>
```

For the definition of table record triggers, a Java class that extends the abstract class `TableTrigger`[API] must be defined inside the `osd:trigger` element. It is advised to define the `annotation/appinfo/osd:trigger` elements just under the element describing the associated table or table type.

Examples:

On a table element:

```
<xs:element name="myTable" type="MyTableType" minOccurs="0" maxOccurs="unbounded">
 <xs:annotation>
  <xs:appinfo>
   <osd:table>
    <primaryKeys>/key</primaryKeys>
   </osd:table>
   <osd:trigger class="com.foo.MyTableTrigger" />
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

On a table type element:

```
<xs:complexType name="MyTableType">
   ...
   <xs:annotation>
    <xs:appinfo>
    <osd:trigger class="com.foo.MyTableTrigger">
    <param1>...</param1>
    <param...n>...</param...n>
    </osd:trigger>
    </xs:appinfo>
   </xs:annotation>
   ...
</xs:complexType>
```

As additional parameters can be defined, the implemented Java class must conform to the JavaBean protocol. In the example above, the Java class must define the methods: `getParam1()`, `setParam1(String)`, `getParamX()`, `setParamX(String)`, etc.

Auto-incremented values

It is possible to define auto-incremented values. Auto-incremented values are only allowed inside tables, and they must be of the type `xs:int` or `xs:integer`.

An auto-increment is specified in the data model using the element `osd:autoIncrement` under the element `annotation/appinfo`.

Example:

```
<xs:element name="autoIncrementedValue" type="xs:int">
 <xs:annotation>
  <xs:appinfo>
   <osd:autoIncrement />
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

Also, there are two optional elements, `start` and `step`:

- The `start` attribute specifies the first value for this auto-increment. If this attribute is not specified, then the value `1` is set by default.
- The `step` attribute specifies the step for the next value to be generated by the auto-increment. If this attribute is not specified, then the value *1* is set by default.

Example:

```
<xs:element name="autoIncrementedValue" type="xs:int">
 <xs:annotation>
  <xs:appinfo>
   <osd:autoIncrement>
    <start>100</start>
    <step>5</step>
   </osd:autoIncrement>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

A field specifying an `osd:autoIncrement` has the following behavior:

- The computation and allocation of the field value are performed whenever a new record is inserted and the field value is undefined.
- No allocation is performed if a programmatic insertion already specifies a non-`null` value. For example, if an archive import or an XML import specifies the value, that value is preserved.

  Consequently, the allocation is not performed for a record insertion in occulting or overwriting modes.
- A newly allocated value is, whenever possible, unique in the scope of the repository. More precisely, the uniqueness of the allocation spans over all the datasets of the data model, and it also spans over all the dataspaces. The latter case allows the merge of a dataspace into its parent with a reasonable guarantee that there will be no conflict if the `osd:autoIncrement` is part of the records' primary key.

  This principle has a very specific limitation: when a mass update transaction that specifies values is performed at the same time as a transaction that allocates a value on the same field, it is possible that the latter transaction will allocate a value that will be set by the first transaction (there is no locking between different dataspaces).

Internally, the auto-increment value is stored in the 'Auto-increments' table of the repository. In the user interface, it can be accessed by administrators in the 'Administration' area. This field is automatically updated so that it defines the greatest value ever set on the associated `osd:autoIncrement` field, in any instance or dataspace in the repository. This value is computed, taking into account the max value found in the table being updated.

In certain cases, for example when multiple environments have to be managed (development, test, production), each with different auto-increment ranges, it may be required to avoid this "max value" check. This particular behavior can be achieved using the `disableMaxTableCheck` property. It is generally not recommended to enable this property unless it is absolutely necessary, as this could generate conflicts in the auto-increment values. However, this property can be set in the following ways:

- Locally, by setting a parameter element in the auto-increment declaration: `<disableMaxTableCheck>true</disableMaxTableCheck>`,
- For the whole data model, by setting `<osd:autoIncrement disableMaxTableCheck="true"/>` in the element `xs:appinfo` of the data model declaration, or
- Globally, by setting the property `ebx.autoIncrement.disableMaxTableCheck=true` in the EBX main configuration file.

  See TIBCO EBX main configuration file [p 369].

> **Note**
>
> When this option is enabled globally, it becomes possible to create records in the table of auto-increments, for example by importing from XML or CSV. If this option is not selected, creating records in the table of auto-increments is prohibited to ensure the integrity of the repository.

CHAPITRE **84**

Labels and messages

TIBCO EBX allows to have custom labels and error messages for data models to be displayed in the interface.

Ce chapitre contient les sections suivantes :

1. Label and description
2. Enumeration labels
3. Mandatory error message (osd:mandatoryErrorMessage)
4. Conversion error message
5. Facet validation message with severity

84.1 Label and description

A label and a description can be added to each node in an adaptation model.

In EBX, each adaptation node is displayed with its label. If no label is defined, the name of the element is used.

Two different notations can be used:

Full	The label and description are defined by the elements `<osd:label>` and `<osd:description>` respectively.
Simple	The label is extracted from the text content, ending at the first period ('.'), with a maximum of 60 characters. The description uses the remainder of the text.

The description may also have a hyperlink, either a standard HTML `href` to an external document, or a link to another node of the adaptation within EBX.

- When using the `href` notation or any other HTML, it must be properly escaped.
- EBX link notation is not escaped and must specify the path of the target, for example:

  ```
  <osd:link path="../misc1">Link to another node in the adaptation</osd:link>
  ```

Example:

```
<xs:element name="misc1" type="xs:string">
 <xs:annotation>
  <xs:documentation>
```

```
   Miscellaneous 1. This is the description of miscellaneous element #1.
   Click <a href="https://www.tibco.com" target="_blank">here</a>
   to learn more.
  </xs:documentation>
 </xs:annotation>
</xs:element>
<xs:element name="misc2" type="xs:string">
 <xs:annotation>
  <xs:documentation>
   <osd:label>
    Miscellaneous 2
   </osd:label>
   <osd:description>
    This is the miscellaneous element #2 and here is a
    <osd:link path="../misc1"> link to another node in the
     adaptation</osd:link>.
   </osd:description>
  </xs:documentation>
 </xs:annotation>
</xs:element>
```

If a node points to a named type, then the label of the node replaces the label of the named type. The same mechanism applies to the description of the node (element `osd:description`).

> **Note**
>
> Regarding whitespace management, the label of a node is always *collapsed* when displayed. That is, contiguous sequences of blanks are collapsed to a single blank, and leading and trailing blanks are removed. In descriptions, however, whitespaces are always *preserved*.

Dynamic labels and descriptions

As an alternative to statically defining the localized labels and descriptions for each node, it is possible to specify a Java class that programmatically determines the labels and descriptions for the nodes of the data model. To define the class, include the element `osd:documentation`, with the attribute `class` in the data model. It is possible to pass JavaBean properties using nested parameter elements.

Example:

```
<xs:schema ...>
 <xs:annotation>
  <xs:appinfo>
   <osd:documentation class="com.foo.MySchemaDocumentation">
    <param1>...</param1>
    <param2>...</param2>
   </osd:documentation>
  </xs:appinfo>
 </xs:annotation>
 ...
</xs:schema ...>
```

The labels and descriptions that are provided programmatically take precedence over the ones defined locally on individual nodes.

Voir aussi *`SchemaDocumentation`*[API]

84.2 Enumeration labels

In an enumeration, a simple, non-localized label can be added to each enumeration element, using the attribute `osd:label`.

> **Attention**
>
> Labels defined for an enumeration element are always `collapsed` when displayed.

Example:

```
<xs:element name="Service" maxOccurs="unbounded">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="1" osd:label="Blue" />
   <xs:enumeration value="2" osd:label="Red" />
   <xs:enumeration value="3" osd:label="White" />
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

It is also possible to fully localize the labels using the standard `xs:documentation` element. If both non-localized and localized labels are added to an enumeration element, the non-localized label will be displayed in any locale that does not have a label defined.

Example:

```
<xs:element name="access" minOccurs="0">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="readOnly">
    <xs:annotation>
     <xs:documentation xml:lang="en-US">
      read only
     </xs:documentation>
     <xs:documentation xml:lang="fr-FR">
      lecture seule
     </xs:documentation>
    </xs:annotation>
   </xs:enumeration>
   <xs:enumeration value="readWrite">
    <xs:annotation>
     <xs:documentation xml:lang="en-US">
      read/write
     </xs:documentation>
     <xs:documentation xml:lang="fr-FR">
      lecture écriture
     </xs:documentation>
    </xs:annotation>
   </xs:enumeration>
   <xs:enumeration value="hidden">
    <xs:annotation>
     <xs:documentation xml:lang="en-US">
      hidden
     </xs:documentation>
     <xs:documentation xml:lang="fr-FR">
      masqué
     </xs:documentation>
    </xs:annotation>
   </xs:enumeration>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

84.3 Mandatory error message (osd:mandatoryErrorMessage)

If the node specifies the attribute `minOccurs="1"` (default behavior), then an error message, which must be provided, is displayed if the user does not complete the field. This error message can be defined specifically for each node using the element `osd:mandatoryErrorMessage`.

Example:

```
<xs:element name="birthDate" type="xs:date">
 <xs:annotation>
  <xs:documentation>
   <osd:mandatoryErrorMessage>
    Please give your birth date.
   </osd:mandatoryErrorMessage>
  </xs:documentation>
 </xs:annotation>
</xs:element>
```

The mandatory error message can be localized:

```
<xs:documentation>
 <osd:mandatoryErrorMessage xml:lang="en-US">
  Name is mandatory
 </osd:mandatoryErrorMessage>
 <osd:mandatoryErrorMessage xml:lang="fr-FR">
  Nom est obligatoire
 </osd:mandatoryErrorMessage>
</xs:documentation>
```

> **Note**
>
> Regarding whitespace management, the enumeration labels are always *collapsed* when displayed.

84.4 Conversion error message

For each predefined XML Schema element, it is possible to define a specific error message if the user entry has an incorrect format.

Example:

```
<xs:element name="email" type="xs:string">
 <xs:annotation>
  <xs:documentation>
   <osd:ConversionErrorMessage xml:lang="en-US">
    Please enter a valid email address.
   </osd:ConversionErrorMessage>
   <osd:ConversionErrorMessage xml:lang="fr-FR">
    Saisissez un e-mail valide.
   </osd:ConversionErrorMessage>
  </xs:documentation>
 </xs:annotation>
</xs:element>
```

84.5 Facet validation message with severity

The validation message that is displayed when the value of a field does not comply with a constraint can define a custom severity, a default non-localized message, and localized message variants. If no severity is specified, the default level is `error`. Blocking constraints *must* have the severity `error`.

XML Schema facet (osd:validation)

The validation message is described by the element `osd:validation` in `annotation/appinfo` under the definition of the facet.

Example:

```
<xs:element name="zipCode">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <!--facet is not localized, but validation message is localized-->
   <xs:minInclusive value="01000">
    <xs:annotation>
     <xs:appinfo>
      <osd:validation>
       <severity>error</severity>
       <message>Non-localized message.</message>
       <message xml:lang="en-US">English error message.</message>
       <message xml:lang="fr-FR">Message d'erreur en français.</message>
      </osd:validation>
     </xs:appinfo>
    </xs:annotation>
   </xs:minInclusive>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

XML Schema enumeration facet (osd:enumerationValidation)

The validation message is described by the element `osd:enumerationValidation` in `annotation/appinfo` under the definition of the field.

Example:

```
<xs:element name="Gender">
 <xs:annotation>
  <xs:appinfo>
   <osd:enumerationValidation>
    <severity>error</severity>
    <message>Non-localized message.</message>
    <message xml:lang="en-US">English error message.</message>
    <message xml:lang="fr-FR">Message d'erreur en français.</message>
   </osd:enumerationValidation>
  </xs:appinfo>
 </xs:annotation>
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="0" osd:label="male" />
   <xs:enumeration value="1" osd:label="female" />
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

EBX facet (osd:validation)

The validation message is described by the element `osd:validation` under the definition of the facet (which is defined in `annotation/appinfo/otherFacets`).

Example:

```
<xs:element name="price" type="xs:decimal">
 <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:minInclusive path="../priceMin">
     <osd:validation>
      <severity>error</severity>
      <message>Non-localized message.</message>
      <message xml:lang="en-US">English error message.</message>
      <message xml:lang="fr-FR">Message d'erreur en français.</message>
     </osd:validation>
    </osd:minInclusive>
   </osd:otherFacets>
  </xs:appinfo>
 </xs:annotation>
</xs:element>
```

CHAPITRE 85

Additional properties

Ce chapitre contient les sections suivantes :

1. Default values
2. Access properties
3. Information
4. Default view
5. Comparison mode
6. Apply last modifications policy
7. Categories

85.1 Default values

In a data model, it is possible to specify a default value for a field using the attribute `default`. This property is used to assign a default value if no value is defined for a field.

The default value is displayed in the user input field at the creation time. That is, the default value will be displayed when creating a new record or adding a new occurrence to an aggregated list.

Example:

In this example, the element specifies a default string value.

```
<xs:element name="fieldWithDefaultValue" type="xs:string" default="aDefaultValue" />
```

85.2 Access properties

The attribute `osd:access` defines the access mode, that is, whether the data of a particular data model node can be read and/or written. This attribute must have one of the following values: `RW`, `R-`, `CC` or `--`.

For each XML Schema node, three types of adaptation are possible:

1. *Adaptation terminal node*

   This node is displayed with an associated value in TIBCO EBX. When accessed using the method `Adaptation.get()`, it uses the adaptation search algorithm.

2. *Adaptation non-terminal node*

This node is a complex type that is only displayed in EBX if it has one child node that is also an adaptation terminal node. It has no value of its own. When accessed using the method `Adaptation.get()`, it returns `null`.

3. *Non-adaptable node*

   This node is not an adaptation terminal node and has no child adaptation terminal nodes. This node is never displayed in EBX. When accessing using the method `Adaptation.get()`, it returns the node default value if one is defined, otherwise it returns `null`.

Voir aussi *`Adaptation`*[API]

Access mode	Behavior
`RW`	*Adaptation terminal node*: value can be read and written in EBX.
`R-`	*Adaptation terminal node*: value can only be read in EBX.
`CC`	*Cut*: This is not an adaptation terminal node and none of its children are adaptation terminal nodes. This "instruction" has priority over any child node regardless of the value of their *access* attribute.
`--`	If the node is a simple type, it is not adaptable. If the node is a complex type, it is not an adaptation terminal node and does not define any child nodes. The `root` node of a data model must specify this access mode.
Default	If the *access* attribute is not defined: • If the node is a computed value, it is considered to be `R-` • If the node is a simple type and its value is not computed, it is considered to be `RW` • If the node is an aggregated list, it is then a terminal value and is considered to be `RW` • Otherwise, it is not an adaptation terminal node and it does not define anything about its child nodes.

Example:

In this example, the element is adaptable because it is an adaptation terminal node.

```
<xs:element name="proxyIpAddress" type="xs:string" osd:access="RW"/>
```

85.3 Information

The element `osd:information` allows specifying additional information. This information can then be used by the integration code, for any purpose, by calling the method `SchemaNode.getInformation`[API].

Example:

```
<xs:element name="misc" type="xs:string">
  <xs:annotation>
    <xs:appinfo>
   <osd:information>
    This is the text information of miscellaneous element.
   </osd:information>
    </xs:appinfo>
```

```
  </xs:annotation>
</xs:element>
```

85.4 Default view

Hiding a field or a table in the default view

It is possible for a table or field inside a table to be hidden by default in EBX by using the element `osd:defaultView/hidden`. This property is used to hide elements from the default view of a dataset without defining specific access permissions. That is, elements hidden by default will not be visible in any default forms and views, whether tabular or hierarchical, generated from the structure of the associated data model.

Attention

- If an element is configured to be hidden in the default view of a dataset, then the access permissions associated with this field will not be evaluated.
- It is possible to display a field that is hidden in the default view of a dataset by defining a view. Only in this case will the access permissions associated with this field be evaluated to determine whether the field will be displayed or not.
- It is not possible to display a table that is hidden in the default view of a dataset (in the navigation pane).

Example:

In this example, the element is hidden in the default view of a dataset.

```
<xs:element name="hiddenField" type="xs:string" minOccurs="0"/>
   <xs:annotation>
  <xs:appinfo>
    <osd:defaultView>
    <hidden>true</hidden>
    </osd:defaultView>
  </xs:appinfo>
   </xs:annotation>
</xs:element>
```

Hiding groups and fields in views

It is possible for a field or a group to be hidden in all views of a table by using the element `osd:defaultView/hiddenInViews`. This property is used to hide elements from the tabular (including the default tabular view) and hierarchical views of a dataset without defining specific access permissions. That is, hidden elements will not be visible in any views, whether tabular or hierarchical, created from the structure of the associated data model. However, hidden elements in views will be displayed in forms.

To specify whether or not to hide an element in all views, use the `osd:defaultView/hiddenInViews="true|false"` element.

If this property is set to `true`, then the element will not be selectable when creating a custom view. As a consequence, the element will not be displayed in all views of a table in a dataset.

If a group is configured as hidden in views, then all the fields nested under this group will not be displayed respectively in the views of the table.

Hiding a field in search tools

To specify whether or not to hide an element in search tools, use the element `osd:defaultView/hiddenInSearch="true|false|textSearchOnly"`.

If this property is set to `true`, then the field will not be selectable in the text and typed search tools of a dataset.

If this property is set to `textSearchOnly`, then the field will not be selectable only in the text search of a dataset but will be selectable in the typed search.

> **Note**
>
> If a group is configured as hidden in search tools or only in the text search, then all the fields nested under this group will not be displayed respectively in the search tools or only in the text search.

Example:

```
<xs:element name="hiddenFieldInSearch" type="xs:string" minOccurs="0"/>
   <xs:annotation>
  <xs:appinfo>
    <osd:defaultView>
    <hiddenInSearch>true</hiddenInSearch>
    </osd:defaultView>
  </xs:appinfo>
   </xs:annotation>
</xs:element>
```

In this example, the element is hidden in the text and typed search tools of a dataset.

```
<xs:element name="hiddenFieldOnlyInTextSearch" type="xs:string" minOccurs="0"/>
   <xs:annotation>
  <xs:appinfo>
    <osd:defaultView>
    <hiddenInSearch>textSearchOnly</hiddenInSearch>
    </osd:defaultView>
  </xs:appinfo>
   </xs:annotation>
</xs:element>
```

In this example, the element is hidden only in the text search tool of a dataset.

Hiding a field in Data Services

To specify whether or not to hide an element in data services, use the element `osd:defaultView/hiddenInDataServices`. For more information, see Disabling fields from data model [p 639].

> **Note**
>
> - If a group is configured as being hidden, then all the fields nested under this group will be considered as hidden by data services.

Example:

```
<xs:element name="hiddenFieldInDataService" type="xs:string" minOccurs="0"/>
   <xs:annotation>
  <xs:appinfo>
    <osd:defaultView>
    <hiddenInDataServices>true</hiddenInDataServices>
    </osd:defaultView>
  </xs:appinfo>
   </xs:annotation>
</xs:element>
```

In this example, the element is hidden in the Data Service select operation.

Defining a view for the combo box selector of a foreign key

It is possible to specify a published view that will be used to display the target table or the hierarchical view of a foreign key for a smoother selection. If a view has been defined, the selector will be displayed in the user interface in the combo box of this foreign key. The definition of a view can be done by using the XML Schema element `osd:defaultView/widget/viewForAdvancedSelection`.

> **Note**
>
> - This property can only be defined on foreign key fields.
> - The published view must be associated with the target table of the foreign key.
> - If the published view does not exist, then the advanced selection is not available in the foreign key field.

Example:

In this example, the name of a published view is defined to display the target table of a foreign key in the advanced selection.

```
<xs:element name="catalog_ref" type="xs:string" minOccurs="0"/>
   <xs:annotation>
  <xs:appinfo>
   <osd:otherFacets>
    <osd:tableRef>
     <tablePath>/root/Catalogs</tablePath>
    </osd:tableRef>
     </osd:otherFacets>
    <osd:defaultView>
    <widget>
     <viewForAdvancedSelection>catalogView</viewForAdvancedSelection>
    </widget>
    </osd:defaultView>
  </xs:appinfo>
   </xs:annotation>
</xs:element>
```

See Combo-box selector [p 60] for more information.

Customizing a default widget

A widget can be defined using the data model assistant. See Default view > Widget [p 60] for more information.

85.5 Comparison mode

The attribute `osd:comparison` can be included on a terminal node element in order to set its comparison mode. This mode controls how differences are detected for the element during comparisons. The possible values for the attribute are:

default	Element is visible during comparisons of its data.
ignored	No changes will be detected when comparing two versions of the content in records or datasets. During a merge, the data values of an ignored element are not merged when contents are updated, even if the values are different. For new content, the values of ignored elements are merged. During an archive import, values of ignored elements are not imported when contents are updated. For new content, the values of ignored elements are imported.

Note

- If a group is configured as being ignored during comparisons, then all the fields nested under this group will also be ignored.
- If a terminal field does not include the attribute `osd:comparison`, then it will be included by default during comparisons.

Restrictions:

- This property cannot be defined on non-terminal fields.
- Primary key fields cannot be ignored during comparison.

Example:

In this example, the first element is explicitly ignored during comparison, the second element is explicitly included.

```
<xs:element name="fieldExplicitlyIgnoredInComparison"
 type="xs:string" minOccurs="0" osd:comparison="ignored"/>
<xs:element name="fieldExplicitlyNotIgnoredInComparison"
 type="xs:string" minOccurs="0" osd:comparison="default"/>
```

85.6 Apply last modifications policy

The attribute `osd:applyLastModification` can be defined on a terminal node element in order to specify if this element must be included or not in the 'apply last modifications' service that can be executed in a table of a dataset.

The possible values for the attribute are:

default	Last modifications can be applied to this element.
ignored	This element is ignored from the apply last modifications service. That is, the last modification that has been performed on this element cannot be applied to other records.

> **Note**
>
> - If a group is configured as being ignored by the 'apply last modifications' service, then all fields nested under this group will also be ignored.
> - If a terminal field does not include the attribute `osd:applyLastModification`, then it will be included by default in the apply last modifications service.
>
> **Restriction:**
>
> - This property cannot be defined on non-terminal fields.

Example:

In this example, the first element is explicitly ignored in the 'apply last modifications' service, the second element is explicitly included.

```
<xs:element name="fieldExplicitlyIgnoredInApplyLastModification"
 type="xs:string" minOccurs="0" osd:applyLastModification="ignored"/>
<xs:element name="fieldExplicitlyNotIgnoredApplyLastModification"
 type="xs:string" minOccurs="0" osd:applyLastModification="default"/>
```

85.7 Categories

Categories can be used for "filtering", by restricting the display of data model elements.

To create a category, add the attribute `osd:category` to a table node in the data model XSD.

Filters on data

In the example below, the attribute `osd:category` is added to the node in order to create a category named *mycategory*.

```
<xs:element name="rebate" osd:category="mycategory">
   <xs:complexType>
  <xs:sequence>
    <xs:element name="label" type="xs:string"/>
    <xs:element name="beginDate" type="xs:date"/>
    <xs:element name="endDate" type="xs:date"/>
    <xs:element name="rate" type="xs:decimal"/>
  </xs:sequence>
   </xs:complexType>
</xs:element>
```

To activate a defined category filter on a dataset in the user interface, select **Actions** > **Categories** > ***<category name>*** from the navigation pane.

Predefined categories

Two categories with localized labels are predefined:

- Hidden

  An instance node, including a table node itself, is hidden in the default view, but can be revealed by selecting **Actions > Categories >** ***[hidden nodes]*** from the navigation pane.

  A table record node is always hidden.

- Constraint (deprecated)

Restriction

Categories do not apply to table record nodes, except the category 'Hidden'.

CHAPITRE **86**

Data services

This chapter details how WSDL operations' names related to a table are defined and managed by TIBCO EBX.

Ce chapitre contient les sections suivantes :

1. Definition
2. Configuration
3. Publication
4. WSDL and table operations
5. Limitations

86.1 Definition

EBX generates a WSDL that complies with the W3C Web Services Description Language 1.1 standard. By default, WSDL operations refer to a table using the last element of the table path. A WSDL operation name is composed of the action name (prefix) and the table name (suffix). It is possible to refer to tables in WSDL operations using unique names instead of the last element of their paths by overriding the suffix operations' names.

Voir aussi*Data services using the Data Model Assistant* *[p 44]*

86.2 Configuration

Embedded data model

WSDL suffix operations' names are embedded in EBX's repository and linked to a publication. That is, when publishing an embedded data model, the list of WSDL suffix operations' names can be defined in the data model definition, under the 'Configuration > Data services' table and managed by EBX.

Packaged data model

WSDL suffix operations' names are defined in a dedicated `XML` document file and must be named as the data model and end with the keyword `_entities`. For instance, if a data model is named `catalog.xsd`, then the XML document containing the configuration of the WSDL operations' names overrided will be named `catalog_entities.xml`. This `XML` document must also be located in the same location as the

data model. The `XML` document is automatically loaded by EBX if a file that matches this pattern is found when compiling a data model.

86.3 Publication

The suffix operations' names are validated at compilation time and contain a list of couples containing `Path` with a unique table name. Checked validation rules are:

- The `path` is not unique,
- The table name contains a syntax error,
- The table name is not unique in the `XML` document.

86.4 WSDL and table operations

WSDL Generator

An additional validation rule has been added: a unicity check is systematically applied to table names. The SOAP operation name is composed of the operation type as a prefix and, by default, of the table name (last step of the table path) as a suffix. A dataset can contain several identical table names but with different paths. It is possible to override table names that are not unique in order to guarantee the unicity.

SOAP operations

When an operation request on table has been invoked from the SOAP connector, the target table is retrieved by priority, the name corresponds to:

1. an overridden table name,
2. the last step of the table path.

Voir aussi*Data services* *[p 620]*

86.5 Limitations

WSDL operations' names are not available with `external` data models.

CHAPITRE 87

Toolbars

This chapter details how toolbars are defined and managed by TIBCO EBX.

Ce chapitre contient les sections suivantes :

1. Definition
2. Using toolbars

87.1 Definition

Toolbars allow to customize the buttons and menus to display when accessing a table view, a hierarchical view, or a record form.

Toolbars can only be created and published using the *Data Model Assistant* and are available only on `embedded` and `packaged` data models.

For embedded data models, toolbars are embedded in EBX's repository and linked to a publication. That is, when publishing an embedded data model, the toolbars defined in the data model are embedded with the publication of the data model and managed by EBX.

For packaged data models, toolbars are defined in a dedicated `XML` document and must be named as the data model and end with the keyword `_toolbars`. For instance, if a data model is named `catalog.xsd` then the XML document containing the definition of the toolbars must be named `catalog_toolbars.xml`. This `XML` document must also be placed in the same location as the data model. The toolbar document is automatically loaded by EBX if a file complying with this pattern is found when compiling a data model.

Voir aussi

Configuring toolbars using the Data Model Assistant *[p 79]*

Using toolbars in data models *[p 521]*

Toolbar API `ToolbarFactory`[API]

87.2 Using toolbars

Toolbars can be used on `tables` and `associations`.

On tables, it is possible to specify the toolbar to display:

- On the top of a tabular view
- On each row of a tabular view

- On the top of a record form
- On the top of a hierarchical view.

On associations, it is possible to specify the toolbar to display:

- On top of the tabular view of the association
- On each row of the tabular view of the association

Voir aussi

Using toolbars *[p 521]*

Associations *[p 525]*

CHAPITRE 88

Workflow model

The workflow offers two types of steps: 'library' or 'specific'.

'Library' is a bean defined in `module.xml` and is reusable. Using the 'library' bean improves the ergonomics: parameters are dynamically displayed in the definition screens.

A 'specific' object is a bean defined only by its class name. In this case, the display is not dynamic.

Ce chapitre contient les sections suivantes :

1. Bean categories
2. Sample of ScriptTask
3. Sample of ScriptTaskBean
4. Samples of UserTask
5. Samples of Condition
6. Sample of ConditionBean
7. Sample of SubWorkflowsInvocationBean
8. Sample of WaitTaskBean
9. Sample of ActionPermissionsOnWorkflow
10. Sample of WorkflowTriggerBean
11. Sample of trigger starting a process instance

88.1 Bean categories

Step	**Library**	Specific
Scripts	ScriptTaskBean	ScriptTask
Conditions	ConditionBean	Condition
User task	UserTask	

88.2 Sample of ScriptTask

Java Code

A script task has to override the method `execute` as in the following example:

```
public class NppScriptTask_CreateWorkingBranch extends ScriptTask
{
    public void executeScript(ScriptTaskContext aContext) throws OperationException
    {
        Repository repository = aContext.getRepository();
        String initialBranchString = aContext.getVariableString("initialBranch");
        AdaptationHome initialBranch = repository.lookupHome(HomeKey.forBranchName(initialBranchString));
        if (initialBranch == null)
            throw OperationException.createError("Null value for initialBranch");

        HomeCreationSpec spec = new HomeCreationSpec();
        spec.setParent(initialBranch);
        spec.setKey(HomeKey.forBranchName("Name"));
        spec.setOwner(Profile.EVERYONE);
        spec.setHomeToCopyPermissionsFrom(initialBranch);
        AdaptationHome newHome = repository.createHome(spec, aContext.getSession());
                //feeds dataContext
        aContext.setVariableString("workingBranch", newHome.getKey().getName());
    }
}
```

Voir aussi*com.orchestranetworks.workflow.ScriptTask* *ScriptTask*[API]

88.3 Sample of ScriptTaskBean

Java Code

A script task bean has to override the method `executeScript` as in the following example:

```
public class ScriptTaskBean_CreateBranch extends ScriptTaskBean
{
    private String initialBranchName;

    private String newBranch;

    public String getInitialBranchName()
    {
        return this.initialBranchName;
    }

    public void setInitialBranchName(String initialBranchName)
    {
        this.initialBranchName = initialBranchName;
    }

    public String getNewBranch()
    {
        return this.newBranch;
    }

    public void setNewBranch(String newBranch)
    {
        this.newBranch = newBranch;
    }

    public void executeScript(ScriptTaskBeanContext aContext) throws OperationException
    {
        final Repository repository = aContext.getRepository();

        String initialBranchName = this.getInitialBranchName();
        final AdaptationHome initialBranch = repository.lookupHome(HomeKey.forBranchName(initialBranchName));
        final HomeCreationSpec spec = new HomeCreationSpec();
        spec.setParent(initialBranch);
        spec.setKey(HomeKey.forBranchName(XsFormats.SINGLETON.formatDateTime(new Date())));
        spec.setOwner(Profile.EVERYONE);
        spec.setHomeToCopyPermissionsFrom(initialBranch);
        final AdaptationHome branchCreate = repository.createHome(spec, aContext.getSession());
```

```
        this.setNewBranch(branchCreate.getKey().getName());
    }
}
```

Voir aussi*com.orchestranetworks.workflow.ScriptTaskBean* `ScriptTaskBean`[API]

Configuration through module.xml

A script task bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.genericScriptTask.ScriptTaskBean_CreateBranch">
            <documentation xml:lang="fr-FR">
                <label>Créer une branche</label>
                <description>
                    Ce script permet de créer une branche
                </description>
            </documentation>
            <documentation xml:lang="en-US">
                <label>Create a branch</label>
                <description>
                    This script creates a branch
                </description>
            </documentation>
            <properties>
                <property name="initialBranchName" input="true">
                    <documentation xml:lang="fr-FR">
                        <label>Branche initiale</label>
                        <description>
                            Nom de la branche initiale.
                        </description>
                    </documentation>
                    <documentation xml:lang="en-US">
                        <label>Initial branch</label>
                        <description>
                            Initial branch name.
                        </description>
                    </documentation>
                </property>
                <property name="newBranch" output="true">
                    <documentation xml:lang="fr-FR">
                        <label>Nouvelle branche</label>
                        <description>
                            Nom de la branche créée
                        </description>
                    </documentation>
                    <documentation xml:lang="en-US">
                        <label>New branch</label>
                        <description>
                            Created branch name.
                        </description>
                    </documentation>
                </property>
            </properties>
        </bean>
    </beans>
</module>
```

88.4 Samples of UserTask

Service declaration via module.xml

A built-in service can be declared in `module.xml` to be used in the user task definition.

```
 <services>
  <service name="ServiceModule">
     <resourcePath>/service.jsp</resourcePath>
     <type>branch</type>
        <documentation xml:lang="fr-FR">
            <label>Workflow service</label>
               <description>
                Ce service permet de ...
               </description>
  </documentation>
           <documentation xml:lang="en-US">
```

```
            <label>Service workflow</label>
               <description>
                The purpose of this service is ...
               </description>
  </documentation>
           <properties>
           <property name="param1" input="true">
                <documentation xml:lang="fr-FR">
                    <label>Param1</label>
                    <description>Param1 ...</description>
                </documentation>
           </property>
           <property name="param2" output="true">
           </property>
       </properties>
 </service>
    <serviceLink serviceName="adaptationService">
        <importFromSchema>
            /WEB-INF/ebx/schema/schema.xsd
        </importFromSchema>
    </serviceLink>
</services>
```

A more complex UserTask

The GUI is quite similar as the example above. The field 'Rule' must be filled to define the class extending the 'UserTask' to invoke.

```
public class NppUserTask_ValidateProduct extends UserTask
{
    public void handleWorkItemCompletion(UserTaskWorkItemCompletionContext context)
        throws OperationException
    {
        if (context.getCompletedWorkItem().isRejected())
        {
            context.setVariableString(NppConstants.VAR_VALIDATION, "KO");
            context.completeUserTask();
        }
        else if (context.checkAllWorkItemMatchStrategy())
        {
            context.setVariableString(NppConstants.VAR_VALIDATION, "OK");
            context.completeUserTask();
        }
    }

    public void handleCreate(UserTaskCreationContext context) throws OperationException
    {
        CreationWorkItemSpec spec = CreationWorkItemSpec.forOfferring(NppConstants.ROLE_PVALIDATOR);
        spec.setNotificationMail("1");
        context.createWorkItem(spec);
        context.setVariableString(NppConstants.VAR_VALIDATION, "validating");
    }
}
```

Voir aussi ***com.orchestranetworks.workflow.UserTask*** `UserTask`[API]

88.5 **Samples of Condition**

Java Code

The method `evaluate` has to be overridden:

```
public class NppCondition_IsValidationOK extends Condition
{
    public boolean evaluateCondition(ConditionContext context) throws OperationException
    {
        String validation = context.getVariableString("validationResult");
        boolean hasError = "KO".equals(validation);
        return !hasError;
    }
}
```

Voir aussi*com.orchestranetworks.workflow.Condition* `Condition`[API]

88.6 Sample of ConditionBean

Java Code

The method `evaluateCondition` has to be overridden as in the following sample:

```
public class ConditionBean_IsBranchValid extends ConditionBean
{
    private String branchName;

    public String getBranchName()
    {
        return this.branchName;
    }

    public void setBranchName(String branchName)
    {
        this.branchName = branchName;
    }

    public boolean evaluateCondition(ConditionBeanContext aContext) throws OperationException
    {
        final Repository repository = aContext.getRepository();
        Severity severityForValidation = Severity.ERROR;
        String branchToTestName = this.getBranchName();
        final AdaptationHome branchToTest = repository.lookupHome(HomeKey.forBranchName(branchToTestName));
        if (branchToTest.getValidationReportsMap(severityForValidation) != null
            && branchToTest.getValidationReportsMap(severityForValidation).size() > 0)
        {
            return false;
        }
        return true;
    }
}
```

Voir aussi*com.orchestranetworks.workflow.ConditionBean* `ConditionBean`[API]

Configuration through module.xml

The condition bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.genericScriptTask.ConditionBean_IsBranchValid">
            <documentation xml:lang="fr-FR">
                <label>Branche valide ?</label>
                <description>
                    Ce script permet de tester si une branche est valide.
                </description>
            </documentation>
            <documentation xml:lang="en-US">
                <label>Branch valid ?</label>
                <description>
                    This script allows to check if a branch is valid.
                </description>
            </documentation>
            <properties>
                <property name="branchName" input="true">
                    <documentation xml:lang="fr-FR">
                        <label>Branche à contrôler</label>
                        <description>
                            Nom de la branche à valider.
                        </description>
                    </documentation>
                    <documentation xml:lang="en-US">
                        <label>Branch to check</label>
                        <description>
                            Branch name to check.
                        </description>
                    </documentation>
                </property>
            </properties>
        </bean>
```

```
    </beans>
</module>
```

88.7 Sample of SubWorkflowsInvocationBean

Java Code

```
public class MySubWorkflowsInvocationBean extends SubWorkflowsInvocationBean
{
 @Override
 public void handleCreateSubWorkflows(SubWorkflowsCreationContext aContext)
  throws OperationException
 {
  final ProcessLauncher subWorkflow1 = aContext.registerSubWorkflow(
   AdaptationName.forName("validateProduct"),
   "validateProduct1");
  subWorkflow1.setLabel(UserMessage.createInfo("Validate the new product by marketing"));
  subWorkflow1.setInputParameter("workingBranch", aContext.getVariableString("workingBranch"));
  subWorkflow1.setInputParameter("code", aContext.getVariableString("code"));
  subWorkflow1.setInputParameter("service", aContext.getVariableString("marketing"));

  final ProcessLauncher subWorkflow2 = aContext.registerSubWorkflow(
   AdaptationName.forName("validateProduct"),
   "validateProduct2");
  subWorkflow2.setLabel(UserMessage.createInfo("Validate the new product by direction"));
  subWorkflow2.setInputParameter("workingBranch", aContext.getVariableString("workingBranch"));
  subWorkflow2.setInputParameter("code", aContext.getVariableString("code"));
  subWorkflow2.setInputParameter("service", aContext.getVariableString("direction"));

  // Conditional launching.
  if (aContext.getVariableString("productType").equals("book"))
  {
   final ProcessLauncher subWorkflow3 = aContext.registerSubWorkflow(
    AdaptationName.forName("generateISBN"),
    "generateISBN");
   subWorkflow3.setLabel(UserMessage.createInfo("Generate ISBN"));
   subWorkflow3.setInputParameter(
    "workingBranch",
    aContext.getVariableString("workingBranch"));
   subWorkflow3.setInputParameter("code", aContext.getVariableString("code"));
  }

  aContext.launchSubWorkflows();
 }
 @Override
 public void handleCompleteAllSubWorkflows(SubWorkflowsCompletionContext aContext)
  throws OperationException
 {
  aContext.getCompletedSubWorkflows();
  final ProcessInstance validateProductMarketing = aContext.getCompletedSubWorkflow("validateProduct1");
  final ProcessInstance validateProductDirection = aContext.getCompletedSubWorkflow("validateProduct2");
  if (aContext.getVariableString("productType").equals("book"))
  {
   final ProcessInstance generateISBN = aContext.getCompletedSubWorkflow("generateISBN");
   aContext.setVariableString("isbn", generateISBN.getDataContext().getVariableString(
    "newCode"));
  }

  if (validateProductMarketing.getDataContext().getVariableString("Accepted").equals("true")
   && validateProductDirection.getDataContext().getVariableString("Accepted").equals(
    "true"))
   aContext.setVariableString("validation", "ok");
 }
}
```

Voir aussi ***com.orchestranetworks.workflow.SubWorkflowsInvocationBean*** *`SubWorkflowsInvocationBean`*[API]

Configuration through module.xml

`SubWorkflowsInvocationBean` bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.test.MySubWorkflowsInvocationBean"/>
    </beans>
```

```
</module>
```

88.8 Sample of WaitTaskBean

Java Code

```
public class MyWaitTaskBean extends WaitTaskBean
{
 @Override
 public void onStart(WaitTaskOnStartContext aContext)
 {
  Map<String, String> params = new HashMap<String, String>();
  params.put("resumeId", aContext.getResumeId());
  myMethod.callWebService(params);
 }

 @Override
 public void onResume(WaitTaskOnResumeContext aContext) throws OperationException
 {
  // Defines a specific mapping.
  aContext.setVariableString("code", aContext.getOutputParameters().get("isbn"));
  aContext.setVariableString("comment", aContext.getOutputParameters().get("isbnComment"));
 }
}
```

Voir aussi*com.orchestranetworks.workflow.WaitTaskBean* *`WaitTaskBean`*[API]

Configuration through module.xml

`WaitTaskBean` bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.test.MyWaitTaskBean"/>
    </beans>
</module>
```

88.9 Sample of ActionPermissionsOnWorkflow

Java Code

```
package com.orchestranetworks.workflow.test;

import com.orchestranetworks.service.*;
import com.orchestranetworks.workflow.*;
import com.orchestranetworks.workflow.ProcessExecutionContext.*;

/**
 */
public class MyDynamicPermissions extends ActionPermissionsOnWorkflow
{

    public ActionPermission getActionPermission(
        WorkflowPermission aWorkflowAction,
        ActionPermissionsOnWorkflowContext aContext)
    {
        if (WorkflowPermission.VIEW.equals(aWorkflowAction)
            || WorkflowPermission.CREATE_PROCESS.equals(aWorkflowAction))
            return ActionPermission.getEnabled();
        return ActionPermission.getDisabled();
    }

}
```

Voir aussi*com.orchestranetworks.workflow.ActionPermissionsOnWorkflow* *`ActionPermissionsOnWorkflow`*[API]

Configuration through module.xml

`ActionPermissionsOnWorkflow` bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.test.MyDynamicPermissions"/>
    </beans>
</module>
```

88.10 **Sample of WorkflowTriggerBean**

Java Code

```
public class MyWorkflowTriggerBean extends WorkflowTriggerBean
{
 @Override
 public void handleAfterProcessInstanceStart(
  WorkflowTriggerAfterProcessInstanceStartContext aContext) throws OperationException
 {
  final DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());
  final MailSpec spec = aContext.createMailSpec();
  spec.notify(NotificationType.TO, "supervisor@mail.com");

  spec.setSubject("[TRIGGER] After process instance start");
  spec.setBody("The workflow '"
   + policy.formatUserMessage(aContext.getProcessInstance().getLabel())
   + "' has been created.");

  spec.sendMail(Locale.US);
 }

 @Override
 public void handleBeforeProcessInstanceTermination(
  WorkflowTriggerBeforeProcessInstanceTerminationContext aContext) throws OperationException
 {
  final DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

  final MailSpec spec = aContext.createMailSpec();
  spec.notify(NotificationType.TO, "supervisor@mail.com");

  spec.setSubject("[TRIGGER] Before process instance termination");
  spec.setBody("The workflow '"
   + policy.formatUserMessage(aContext.getProcessInstance().getLabel())
   + "' has been completed. The created product is: '"
   + aContext.getVariableString(NppConstants.VAR_CODE) + "'.");

  spec.sendMail(Locale.US);
 }

 @Override
 public void handleAfterWorkItemCreation(WorkflowTriggerAfterWorkItemCreationContext aContext)
  throws OperationException
 {
  DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

  MailSpec spec = aContext.createMailSpec();
  spec.notify(NotificationType.TO, "supervisor@mail.com");

  spec.setSubject("[TRIGGER] After work item creation");
  WorkItem workItem = aContext.getWorkItem();
  State state = workItem.getState();
  String body = "The work item '" + policy.formatUserMessage(workItem.getLabel())
   + "' has been created. \n The step id is : " + aContext.getCurrentStepId()
   + ". \n The work item is in state : " + policy.formatUserMessage(state.getLabel());

  if (workItem.getOfferedTo() != null)
   body += "\n The role is :" + workItem.getOfferedTo().format();
  if (workItem.getUserReference() != null)
   body += "\n The user is :" + workItem.getUserReference().format();

  spec.setBody(body);

  spec.sendMail(Locale.US);
 }

 @Override
```

```
public void handleBeforeWorkItemStart(WorkflowTriggerBeforeWorkItemStartContext aContext)
 throws OperationException
{
 DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

 MailSpec spec = aContext.createMailSpec();
 spec.notify(NotificationType.TO, "supervisor@mail.com");

 spec.setSubject("[TRIGGER] Before work item start");
 spec.setBody("The work item '"
  + policy.formatUserMessage(aContext.getWorkItem().getLabel())
  + "' has been started. \n  The current step id is : "
  + aContext.getCurrentStepId()
  + ". \n The work item user is: '"
  + DirectoryHandler.getInstance(aContext.getRepository()).displayUser(
   aContext.getWorkItem().getUserReference(),
   aContext.getSession().getLocale()) + "'.");

 spec.sendMail(Locale.US);
}

@Override
public void handleBeforeWorkItemAllocation(
 WorkflowTriggerBeforeWorkItemAllocationContext aContext) throws OperationException
{
 DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

 MailSpec spec = aContext.createMailSpec();
 spec.notify(NotificationType.TO, "supervisor@mail.com");

 spec.setSubject("[TRIGGER] Before work item allocation");
 spec.setBody("The work item '"
  + policy.formatUserMessage(aContext.getWorkItem().getLabel())
  + "' has been allocated. \n  The current step id is: "
  + aContext.getCurrentStepId()
  + ". \n  The work item user is: '"
  + DirectoryHandler.getInstance(aContext.getRepository()).displayUser(
   aContext.getUserReference(),
   aContext.getSession().getLocale()) + "'.");

 spec.sendMail(Locale.US);
}

@Override
public void handleBeforeWorkItemDeallocation(
 WorkflowTriggerBeforeWorkItemDeallocationContext aContext) throws OperationException
{
 DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

 MailSpec spec = aContext.createMailSpec();
 spec.notify(NotificationType.TO, "supervisor@mail.com");

 spec.setSubject("[TRIGGER] Before work item deallocation");
 spec.setBody("The work item '"
  + policy.formatUserMessage(aContext.getWorkItem().getLabel())
  + "' has been deallocated. \n  The current step id is: "
  + aContext.getCurrentStepId()
  + ". \n  The old work item user is: '"
  + DirectoryHandler.getInstance(aContext.getRepository()).displayUser(
   aContext.getWorkItem().getUserReference(),
   aContext.getSession().getLocale()) + ".");

 spec.sendMail(Locale.US);

}

@Override
public void handleBeforeWorkItemReallocation(
 WorkflowTriggerBeforeWorkItemReallocationContext aContext) throws OperationException
{
 DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

 MailSpec spec = aContext.createMailSpec();
 spec.notify(NotificationType.TO, "supervisor@mail.com");

 spec.setSubject("[TRIGGER] Before work item reallocation");
 spec.setBody("The work item '"
  + policy.formatUserMessage(aContext.getWorkItem().getLabel())
  + "' has been reallocated. \n  The current step id is: "
  + aContext.getCurrentStepId()
  + ". \n  The work item user is: '"
  + DirectoryHandler.getInstance(aContext.getRepository()).displayUser(
   aContext.getUserReference(),
   aContext.getSession().getLocale())
  + "'. The old work item user is: '"
```

```
   + DirectoryHandler.getInstance(aContext.getRepository()).displayUser(
    aContext.getWorkItem().getUserReference(),
    aContext.getSession().getLocale()) + "'.");

  spec.sendMail(Locale.US);

 }
 @Override
 public void handleBeforeWorkItemTermination(
  WorkflowTriggerBeforeWorkItemTerminationContext aContext) throws OperationException
 {
  DisplayPolicy policy = DisplayPolicyFactory.getPolicyForSession(aContext.getSession());

  MailSpec spec = aContext.createMailSpec();
  spec.notify(NotificationType.TO, "supervisor@mail.com");

  spec.setSubject("[TRIGGER] Before work item termination");
  spec.setBody("The work item '"
   + policy.formatUserMessage(aContext.getWorkItem().getLabel())
   + "' has been terminated. \n  The current step id is: " + aContext.getCurrentStepId()
   + ". \n  The work item has been accepted ? " + aContext.isAccepted());

  spec.sendMail(Locale.US);
 }
}
```

Voir aussi*com.orchestranetworks.workflow.WorkflowTriggerBean* *`WorkflowTriggerBean`*[API]

Configuration through module.xml

`WorkflowTriggerBean` bean must be declared in `module.xml`:

```
<module>
    <beans>
        <bean className="com.orchestranetworks.workflow.test.MyWorkflowTriggerBean"/>
    </beans>
</module>
```

88.11 Sample of trigger starting a process instance

Sample

```
public class TriggerWorkflow extends TableTrigger
{
    public void handleAfterModify(AfterModifyOccurrenceContext aContext) throws OperationException
    {
        ValueContext currentRecord = aContext.getOccurrenceContext();
        String code = (String) currentRecord.getValue(Path.parse("/code"));

        //Get published process
        PublishedProcessKey processPublishedKey = PublishedProcessKey.forName("productProcess");
                //Defines process instance
            ProcessLauncher launcher = ProcessLauncherHelper.createLauncher(
            processPublishedKey,
            aContext.getProcedureContext());
        //initialize Data Context
        launcher.setInputParameter("code", "/root/Client[./code=\"" + code + "\"]");
        launcher.setInputParameter("workingBranch", aContext.getAdaptationHome().getKey().getName());

        //Starts process
        launcher.launchProcess();

    }
      //...
}
```

User interface

CHAPITRE **89**

Interface customization

The TIBCO EBX graphical interface can be customized through various EBX APIs.

Ce chapitre contient les sections suivantes :

1. How to embed a Web Component
2. User services
3. Form layout
4. Custom widgets
5. Table filter
6. Record label
7. CSS and JavaScript

89.1 How to embed a Web Component

EBX can be integrated into any application that is accessible through a supported web browser, thanks to the Web Component API.

To embed all or part of EBX in a web page, the HTML tag `<iframe>` should be used by indicating the URL to EBX. This URL can be specified either manually or by using the `UIHttpManagerComponent` API. A single web page may include several iframes that integrate EBX. It is then possible to create a portal made of tables, forms, hierarchical views, etc., from EBX.

Voir aussi*Utiliser TIBCO EBX comme composant web* *[p 211]*

89.2 User services

A user service is an extension of EBX that provides a graphical user interface (GUI) that allows users to access specific or advanced functions.

Powerful custom user services can be developed using the same visual components and data validation mechanisms as standard EBX user interfaces.

Voir aussi*User service overview* *[p 587]*

89.3 Form layout

It is possible to override the default layout of forms in the user interface by highly customizing it with the `UIForm` API. This API provides the standard input components from EBX, which give the possibility to customize the layout of a form, while having the same components and standard behavior as record forms using the default layout.

Voir aussi

`UIForm`[API]

`UIFormPaneWriter`[API]

`UIWidget`[API]

`UIFormHeader`[API]

`UIFormBottomBar`[API]

89.4 Custom widgets

Custom widgets are included in the Java API to allow the development of user interface components for fields or groups of fields. A custom widget (`UICustomWidget`) allows, for a given node, to control the area where the input or display component is located. This allows having an input and display component that is fully customizable in HTML. The standard components (`UIWidgets`) are available and can be used. The custom widget can implement several display aspects: input component in the form, display in the form, display in a table cell. If a custom widget writes its own HTML components, it has the possibility to save the value in the database when submitting the form.

Voir aussi*`UICustomWidget`*[API]

89.5 Table filter

A table filter allows, for a given table, to create a criteria input form in order to apply a filter to the table view. The `UITableFilter` API is used to implement a table filter with a custom UI. It provides methods to create a UI that automatically adapts to the underlying data format (for example, by displaying a combo box when applicable).

Voir aussi

`UITableFilter`[API]

Proprits des lments du modle de donnes [p 55]

`UILabelRendererForHierarchy`[API]

89.6 Record label

EBX uses a label to display a reference to a given record (for example a foreign key). Labels are also used in the title of a record form and in hierarchical views. This label can be customized in the model using expressions. It is also possible to customize labels using the `UILabelRenderer` API.

Voir aussi*`UILabelRenderer`*[API]

89.7 CSS and JavaScript

It is possible to integrate CSS and JavaScript files in each EBX page by declaring them in the registration module. The inclusion of JavaScript files can be subject to conditions through development depending on the context.

Voir aussi

Module registration *[p 484]*

Development recommendations *[p 615]*

`UIDependencyRegisterer`[API]

CHAPITRE 90

Overview

A user service is an extension to TIBCO EBX that provides a graphical user interface (GUI) allowing users to access specific or advanced functionalities.

An API is available allowing the development of powerful custom user services using the same visual components and data validation mechanisms as standard EBX user interfaces.

Ce chapitre contient les sections suivantes :

1. Nature
2. Declaration
3. Display
4. Legacy user services

90.1 **Nature**

User services exist in different types called *natures*. The nature defines the minimal elements (dataspace, dataset, table, record...) that need to be selected to execute the service. The following table lists the available natures.

Nature	Description
`Dataspace`	The nature of a user service that can be launched from the actions menu of a dataspace (branch or snapshot) or from any context where the current selection implies selecting a dataspace.
`Dataset`	The nature of a user service that can be launched from the actions menu of a dataset or from any context where the current selection implies selecting a dataset.
`TableView`	The nature of a user service that can be launched from the toolbar of a table, regardless of the selected view, or from any context where the current selection implies selecting a table.
`Record`	The nature of a user service that can be launched from the toolbar of a record form or from any context where the current selection implies selecting a *single* record.
`Hierarchy`	The nature of a user service that can be launched from the toolbar of a table when a hierarchy view is selected.
`HierarchyNode`	The nature of a user service that can be launched from the menu of a table hierarchy view node. Currently, only *record* hierarchy nodes are supported.
`Association`	The nature of a user service that can be launched from the target table view of an association or for any context where the current selection implies selecting the target table view of an association.
`AssociationRecord`	The nature of a user service that can be launched from the form of a target record of an association node or from any context where the current selection implies selecting a *single* association target record.

90.2 **Declaration**

A user service can be declared at two levels:

- Module,
- Data model.

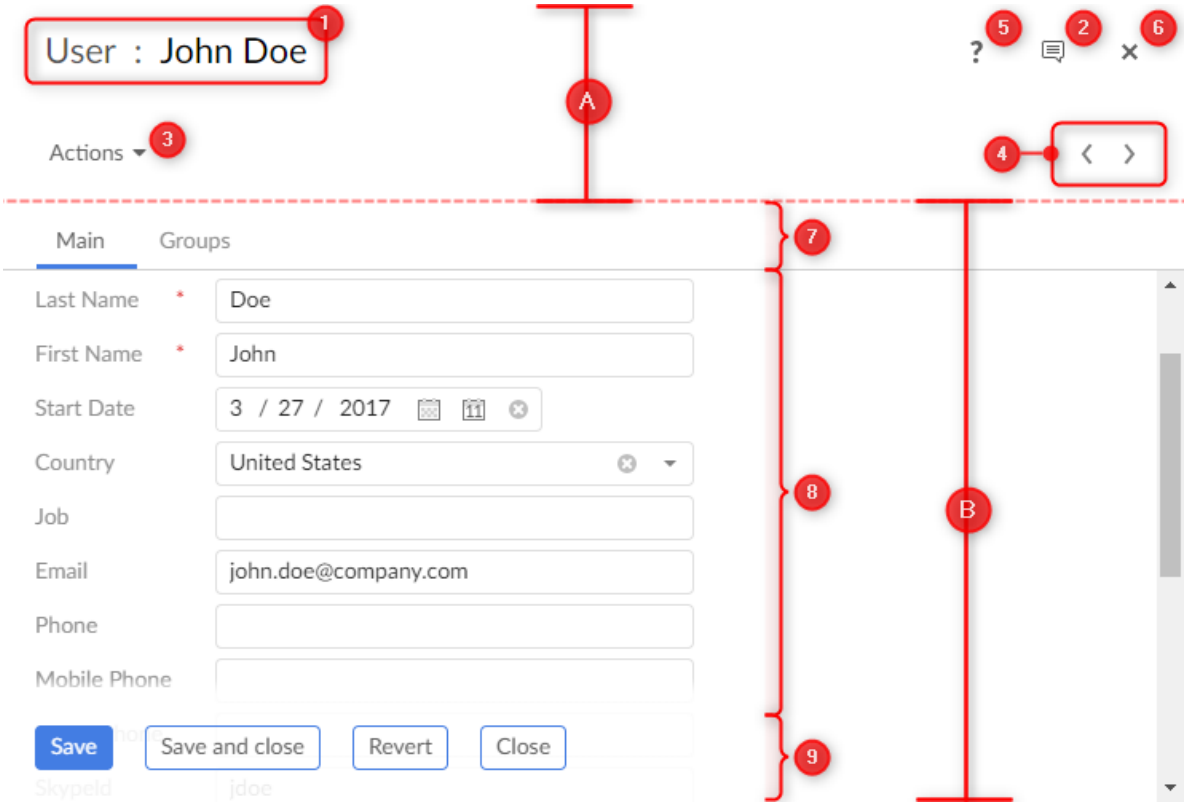
A service declared by a data model can only be launched when the current selection includes a dataset of this model. The user service cannot be of the `Dataspace` nature.

A service declared by a module may be launched for any dataspace or dataset.

The declaration can add restrictions on selections that are valid for the user service.

90.3 Display

On the following figure are displayed the functional areas of a user service.



A. Header B. Form	1. Breadcrumb 2. Message box button 3. Top toolbar 4. Navigation buttons 5. Help button 6. Close button (pop-ups only) 7. Tabs 8. Form pane (one per tab) 9. Bottom buttons

Most areas are optional and customizable. Refer to Quick start [p 591], Implementing a user service [p 595] and Declaring a user service [p 609] for more details.

90.4 Legacy user services

Before the 5.8.0 version, user services were declared in XML and based on Servlet/JSP. Although this type of declaration should no longer be used, the *legacy documentation* is still available.

CHAPITRE **91**

Quick start

Ce chapitre contient les sections suivantes :

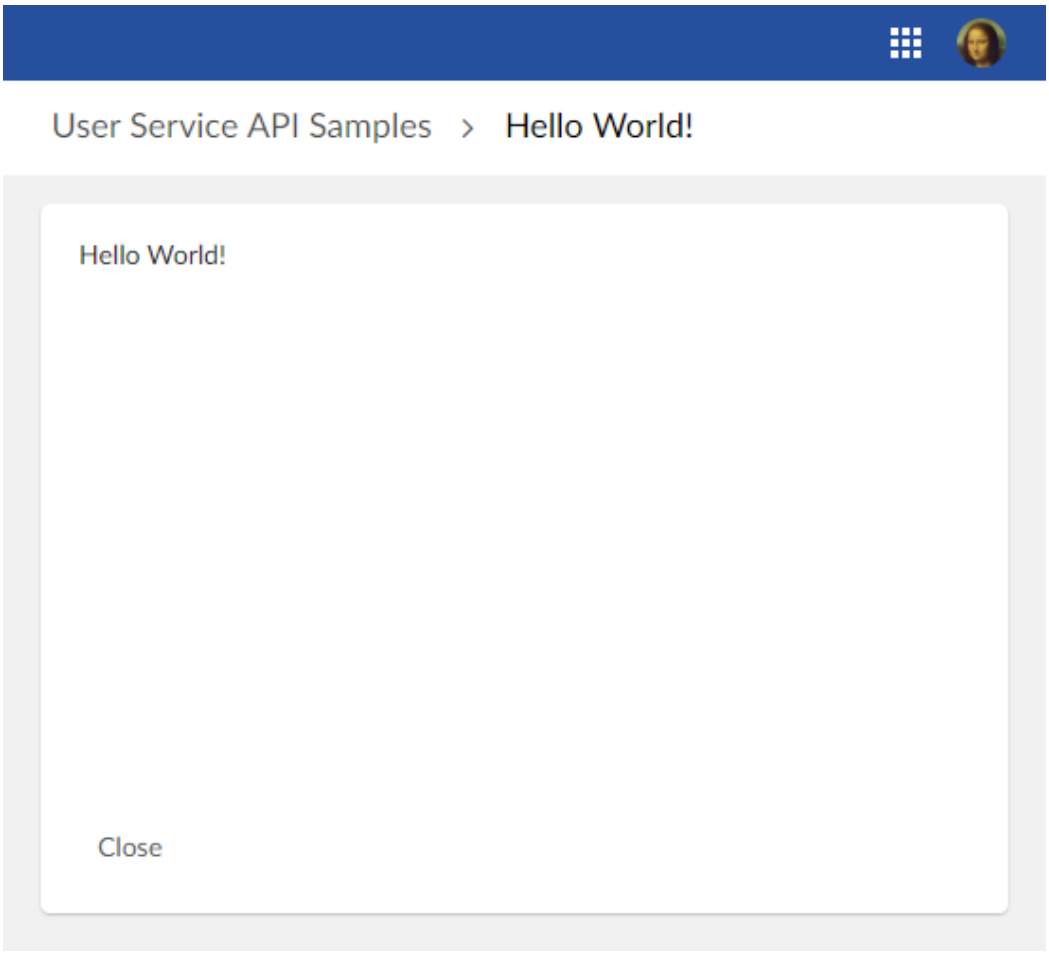
1. Main classes
2. Hello world

91.1 Main classes

The minimum requirement is to implement two classes, one for the service declaration and one for the implementation itself.

91.2 **Hello world**

The sample is a dataset user service that simply displays a "hello" message, it can be launched from the action menu of a dataset:



The service implementation class must implement the interface `UserService<DatasetEntitySelection>`:

```
/**
 * This service displays hello world!
 */
public class HelloWordService implements UserService<DatasetEntitySelection>
{
 public HelloWordService()
 {
 }

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DatasetEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  // Set bottom bar
  UIButtonSpecNavigation closeButton = aConfigurator.newCloseButton();
  closeButton.setDefaultButton(true);
  aConfigurator.setLeftButtons(closeButton);

  // Set content callback
  aConfigurator.setContent(this::writeHelloWorld);
 }

 private void writeHelloWorld(
  UserServicePaneContext aPaneContext,
```

```
  UserServicePaneWriter aWriter)
 {
  // Display Hello World!

  aWriter.add("<div ");
  aWriter.addSafeAttribute("class", UICSSClasses.CONTAINER_WITH_TEXT_PADDING);
  aWriter.add(">");
  aWriter.add("Hello World!");
  aWriter.add("</div>");
 }

 @Override
 public void setupObjectContext(
  UserServiceSetupObjectContext<DatasetEntitySelection> aContext,
  UserServiceObjectContextBuilder aBuilder)
 {
  // No context yet.
 }

 @Override
 public void validate(UserServiceValidateContext<DatasetEntitySelection> aContext)
 {
  // No custom validation is necessary.
 }

 @Override
 public UserServiceEventOutcome processEventOutcome(
  UserServiceProcessEventOutcomeContext<DatasetEntitySelection> aContext,
  UserServiceEventOutcome anEventOutcome)
 {
  // By default do not modify the outcome.
  return anEventOutcome;
 }
}
```

The declaration class must implement the interface `UserServiceDeclaration.OnDataset`:

```
/**
 * Declaration for service hello world!
 */
public class HelloWorldServiceDeclaration implements UserServiceDeclaration.OnDataset
{
 // The service key identifies the user service.
 private static final ServiceKey serviceKey = ServiceKey.forName("HelloWorld");

 public HelloWorldServiceDeclaration()
 {
 }

 @Override
 public ServiceKey getServiceKey()
 {
  return serviceKey;
 }

 @Override
 public UserService<DatasetEntitySelection> createUserService()
 {
  // Creates an instance of the user service.
  return new HelloWordService();
 }

 @Override
 public void defineActivation(ActivationContextOnDataset aContext)
 {
  // The service is activated for all datasets instanciated with
  // the associated data model (see next example).
 }

 @Override
 public void defineProperties(UserServicePropertiesDefinitionContext aContext)
 {
  // This label is displayed in menus that can execute the user service.
  aContext.setLabel("Hello World Service");
 }

 @Override
 public void declareWebComponent(WebComponentDeclarationContext aContext)
 {
 }
}
```

In this sample, the user service is registered by a data model. The data model needs to define a schema extension that implements the following code:

```
public class CustomSchemaExtensions implements SchemaExtensions
{
 @Override
 public void defineExtensions(SchemaExtensionsContext aContext)
 {
  // Register the service.
  aContext.registerUserService(new HelloWorldServiceDeclaration());
 }
}
```

For details on the declaration of schema extensions, see `SchemaExtensions`[API].

CHAPITRE 92

Implementing a user service

Ce chapitre contient les sections suivantes :

1. Implementation interface
2. Life cycle and threading model
3. Object Context
4. Display setup
5. Database updates
6. Ajax
7. REST data services
8. File upload
9. File download
10. User service without display

92.1 Implementation interface

The following table lists, per nature, the interface to implement:

Nature	Interface
`Dataspace`	`UserService<DataspaceEntitySelection>`
`Dataset`	`UserService<DatasetEntitySelection>`
`TableView`	`UserService<TableViewEntitySelection>`
`Record`	`UserService<RecordEntitySelection>`
`Hierarchy`	`UserService<HierarchyEntitySelection>`
`HierarchyNode`	`UserService<HierarchyNodeEntitySelection>`
`Association`	`UserService<AssociationEntitySelection>`
`AssociationRecord`	`UserService<AssociationRecordEntitySelection>`

92.2 Life cycle and threading model

The user service implementation class is:

- Instantiated at the first HTTP request by a call to its declaration **createUserService()** `UserServiceDeclaration.createUserService`[API] method.
- Discarded when the current page goes out of scope or when the session times out.

Access to this class is synchronized by TIBCO EBX to make sure that only one HTTP request is processed at a time. Therefore, the class does not need to be thread-safe.

The user service may have attributes. The state of these attributes will be preserved between HTTP requests. However, developers must be aware that these attributes should have moderate use of resources, such as memory, not to overload the EBX server.

92.3 Object Context

The object context is a container for objects managed by the user service. This context is initialized and modified by the user service's implementation of the method `UserService.setupObjectContext`[API].

An object of the object context is identified by an object key:

```
ObjectKey customerKey = ObjectKey.forName("customer");
```

An object can be:

- A record,
- A dataset,
- A new record not yet persisted,
- A dynamic object.

The object context is maintained between HTTP requests and usually only needs to be set up upon the first request.

Once persisted, a *new record* object is automatically changed to a *plain record* object.

As with **adaptations** `Adaptation`[API], **path** `Path`[API] expressions are used to reference a sub-element of an object.

In the following sample, a pane writer adds a form input mapped to the attribute of an object:

```
// Add an input field for customer's last name.
aWriter.setCurrentObject(customerKey);
aWriter.addFormRow(Path.parse("lastName"));
```

In the following sample, an event callback gets the value of the attribute of an object:

```
// Get value of customer's last name.
ValueContext customerValueContext = aValueContext.getValueContext(customerKey);
String lastName = customerValueContext.getValue(Path.parse("lastName"));
```

A *dynamic object* is an object whose schema is defined by the user service itself. An API is provided to define the schema programmatically. This API allows defining only instance elements (instance nodes). Defining tables is not supported. It supports most other features available with standard EBX data models, such as types, labels, custom widgets, enumerations and constraints, including programmatic ones.

The following sample defines two objects having the same schema:

```
public class SampleService implements UserService<TableViewEntitySelection>
{
 // Define an object key per object:
 private final static ObjectKey _PersonObjectKey = ObjectKey.forName("person");
 private final static ObjectKey _PartnerObjectKey = ObjectKey.forName("partner");

 // Define a path for each property:
 private final static Path _FirstName = Path.parse("firstName");
 private final static Path _LastName = Path.parse("lastName");
 private final static Path _BirthDate = Path.parse("birthDate");

 ...

 // Define and register objects:
 @Override
 public void setupObjectContext(
  UserServiceSetupObjectContext<DataspaceEntitySelection> aContext,
  UserServiceObjectContextBuilder aBuilder)
 {
  if (aContext.isInitialDisplay())
  {
   BeanDefinition def = aBuilder.createBeanDefinition();

   BeanElement firstName = def.createElement(_FirstName, SchemaTypeName.XS_STRING);
   firstName.setLabel("First name");
   firstName.setDescription("This is the given name");
   firstName.setMinOccurs(1);

   BeanElement lastName = def.createElement(_LastName, SchemaTypeName.XS_STRING);
   lastName.setLabel("Last name");
   lastName.setDescription("This is the familly name");
   lastName.setMinOccurs(1);

   BeanElement birthDate = def.createElement(_BirthDate, SchemaTypeName.XS_DATE);
   birthDate.setLabel("Birth date");
   birthDate.addFacetMax(new Date(), false);

   aBuilder.registerBean(_PersonObjectKey, def);
   aBuilder.registerBean(_PartnerObjectKey, def);
  }

  ...
 }
```

92.4 **Display setup**

The display is set up by the user service's implementation of the method `UserService.setupDisplay`[API].

This method is called at each request and can set the following:

- The title (the default is the label specified by the user service declaration),
- The contextual help URL,
- The breadcrumbs,
- The toolbar,
- The bottom buttons.

If necessary, the header and the bottom buttons can be hidden.

The display setup is not persisted and, at each HTTP request, is reset to default before calling the method `UserService.setupDisplay`[API].

Bottom buttons

Buttons may be of two types: *action* and *submit*.

An *action* button triggers an *action* event without submitting the form. By default, the user needs to acknowledge that, by leaving the page, the last changes will be lost. This behavior can be customized.

A *submit* button triggers a *submit* event that always submits the form.

More information on events can be found in the following sections.

Content callback

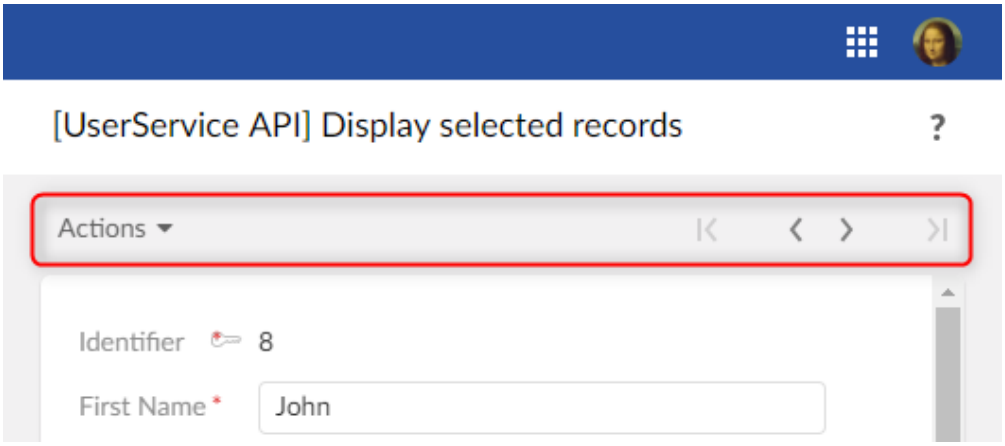
This callback usually implements the interface `UserServicePane`[API] to render a plain EBX form. The callback can also be an instance of `UserServiceRootTabbedPane`[API] to render an EBX form with tabs.

For specific cases, the callback can implement `UserServiceRawPane`[API]. This interface has restrictions but is useful when one wants to implement an HTML form that is not managed by EBX.

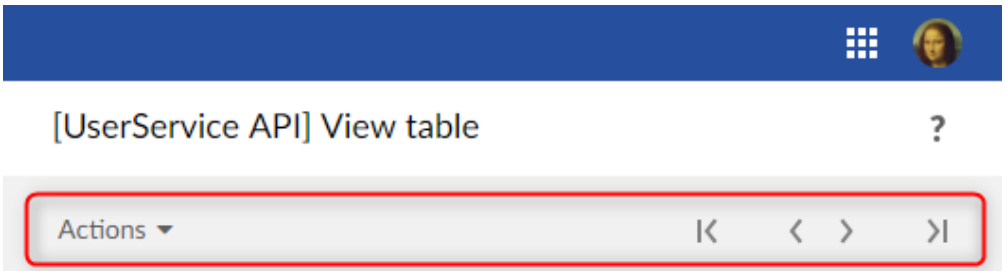
Toolbars

Toolbars are optional and come in two flavors.

The *form* style:



The *table view* style:



The style is automatically selected: toolbars defined for a *record* are of the form style and toolbars defined for a *table* are of the table view style.

Samples

The following sample implements a button that closes the current user service and redirects the user back to the current selection, only if saving the data was successful:

```
public class SampleService implements UserService<...>
{
 private final static ObjectKey _RecordObjectKey = ObjectKey.forName("record");

 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<RecordEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  ...
  // Define a "save and close" button with callback onSave().
  aConfigurator.setLeftButtons(aConfigurator.newSaveCloseButton(this::onSave));
 }

 private UserServiceEventOutcome onSave(UserServiceEventContext anEventContext)
```

```
 {
  ProcedureResult result = anEventContext.save(_RecordObjectKey);
  if (result.hasFailed())
  {
   // Save has failed. Redisplay the user message.
   return null;
  }

  // Save has succeded.Close the service.
  return UserServiceNext.nextClose();
 }
}
```

The following sample is compatible with the Java 6 syntax. Only differences with the previous code are shown:

```
public class SampleService implements UserService<...>
{
 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<RecordEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  ...
  // Define a "save and close" button with callback onSave().
  aConfigurator.setLeftButtons(aConfigurator.newSaveCloseButton(new UserServiceEvent() {
   @Override
   public UserServiceEventOutcome processEvent(UserServiceEventContext anEventContext)
   {
    return onSave(anEventContext);
   }
  }));
 }
}
```

The following sample implements a URL that closes the service and redirects the current user to another user service:

```
public class SampleService implements UserService<...>
{
 ...
 private void writePane(UserServicePaneContext aPaneContext, UserServicePaneWriter aWriter)
 {
  // Displays an ULR that redirect current user.
  String url = aWriter.getURLForAction(this::goElsewhere);
  aWriter.add("<a ");
  aWriter.addSafeAttribute("href", url);
  aWriter.add(">Go elsewhere</a");
 }

 private UserServiceEventOutcome goElsewhere(UserServiceEventContext anEventContext)
 {
  // Redirects current user to another user service.
  ServiceKey serviceKey = ServiceKey.forModuleServiceName("CustomerModule", "CustomService");
   return UserServiceNext.nextService(serviceKey);
 }
}
```

The following code is an implementation of the method `UserService.processEventOutcome`[API], sufficient for simple user services:

```
public class HelloWordService implements UserService<...>
{
 @Override
 public UserServiceEventOutcome processEventOutcome(
  UserServiceProcessEventOutcomeContext<DatasetEntitySelection> aContext,
  UserServiceEventOutcome anEventOutcome)
 {
  // By default do not modify the outcome.
  return anEventOutcome;
 }
}
```

The following sample is a more complex "wizard" service that includes three steps, each having its own `UserService.setupDisplay`[API] method:

```
// Custom outcome values.
```

```
public enum CustomOutcome implements UserServiceEventOutcome {
 displayStep1, displayStep2, displayStep3
};

// All steps of the  wizard service implement this interface.
public interface WizardStep
{
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator);
}

// The user service implementation.
public class WizardService implements UserService<...>
{
 // Attribute for current step.
 private WizardStep step = new WizardStep1();

 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  ...

  // Display current step.
  this.step.setupDisplay(aContext, aConfigurator);
 }

 @Override
 public UserServiceEventOutcome processEventOutcome(
  UserServiceProcessEventOutcomeContext<DataspaceEntitySelection> aContext,
  UserServiceEventOutcome anEventOutcome)
 {
  // Custom outcome value processing.

  if (anEventOutcome instanceof CustomOutcome)
  {
   CustomOutcome action = (CustomOutcome) anEventOutcome;
   switch (action)
   {
   case displayStep1:
    this.step = new WizardStep1();
    break;

   case displayStep2:
    this.step = new WizardStep2();
    break;

   case displayStep3:
    this.step = new WizardStep3();
    break;
   }

   // Redisplay the user service.
   return null;
  }

  // Let EBX® process the event outcome.
  return anEventOutcome;
 }
}
```

92.5 **Database updates**

An event callback may update the database.

The following sample saves two objects using a single transaction:

```
public class MultipleObjectsSampleService implements UserService<...>
{
 // This service defines a two objects having same schema.
 private final static ObjectKey _Person1_ObjectKey = ObjectKey.forName("person1");
 private final static ObjectKey _Person2_ObjectKey = ObjectKey.forName("person2");

 ...

 // Save button callback.
```

```
 private UserServiceEventOutcome onSave(UserServiceEventContext aContext)
 {
  ProcedureResult result = aContext.save(_Person1_ObjectKey, _Person2_ObjectKey);
  if (result.hasFailed())
  {
   //Save failed. Redisplay the service.
   //The user interface will automatically report error messages.
   return null;
  }

  // Save succeeded. Close the service.
  return UserServiceNext.nextClose();
 }
}
```

The following sample updates the database using a **procedure** `Procedure`[API]:

```
import com.orchestranetworks.service.*;
import com.orchestranetworks.userservice.*;

public class MultipleObjectsSampleService implements UserService<...>
{
 ...

 // Event callback.
 private UserServiceEventOutcome onUpdateSomething(UserServiceEventContext aContext)
 {
  Procedure procedure = new Procedure()
  {
   public void execute(ProcedureContext aContext) throws Exception
   {
    // Code that updates database should be here.
    ...
   }
  };

  UserServiceTransaction transaction = aContext.createTransaction();
  transaction.add(procedure);

  ProcedureResult result = transaction.execute();
  if (result.hasFailed())
  {
   aContext.addError("Procedure failed");
  }
  else
  {
   aContext.addInfo("Procedure succeeded");
  }

  return null;
}
```

92.6 **Ajax**

A user service can implement Ajax callbacks. An Ajax callback must implement the interface `UserServiceAjaxRequest`[API].

The client calls an Ajax callback using the URL generated by: `UserServiceResourceLocator.getURLForAjaxRequest`[API].

To facilitate the use of Ajax components, EBX provides the JavaScript prototype `EBX_AJAXResponseHandler` for sending the request and handling the response. For more information on `EBX_AJAXResponseHandler` see `UserServiceAjaxRequest`[API].

The following sample implements an Ajax callback that returns partial HTML:

```
public class AjaxSampleService implements UserService<DataspaceEntitySelection>
{
 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  aConfigurator.setLeftButtons(aConfigurator.newCloseButton());
  aConfigurator.setContent(this::writePane);
```

```
 }

 /**
  * Displays an URL that will execute the callback
  * and display the returned partial HTML inside a <div> tag.
  */
 private void writePane(UserServicePaneContext aPaneContext, UserServicePaneWriter aWriter)
 {
  // Generate the URL of the Ajax callback.
  String url = aWriter.getURLForAjaxRequest(this::ajaxCallback);

  // The id of the <div> that will display the partial HTML returned by the Ajax callback.
  String divId = "sampleId";

  aWriter.add("<div ");
  aWriter.addSafeAttribute("class", UICSSClasses.CONTAINER_WITH_TEXT_PADDING);
  aWriter.add(">");

  // Display the URL that will execute the callback.
  aWriter.add("<a ");
  aWriter.addSafeAttribute("href", "javascript:sample_sendAjaxRequest('" + url + "', '"
   + divId + "')");
  aWriter.add(">");
  aWriter.add("Click to call a user service Ajax callback");
  aWriter.add("</a>");

  // Output the <div> tag that will display the partial HTML returned by the callback.
  aWriter.add("<div ");
  aWriter.addSafeAttribute("id", divId);
  aWriter.add("></div>");

  aWriter.add("</div>");

  // JavaScript method that will send the Java request.
  aWriter.addJS_cr();
  aWriter.addJS_cr("function sample_sendAjaxRequest(url, targetDivId) {");
  aWriter.addJS_cr("  var ajaxHandler = new EBX_AJAXResponseHandler();");

  aWriter.addJS_cr("  ajaxHandler.handleAjaxResponseSuccess = function(responseContent) {");
  aWriter.addJS_cr("    var element = document.getElementById(targetDivId);");
  aWriter.addJS_cr("    element.innerHTML = responseContent;");
  aWriter.addJS_cr("  };");

  aWriter.addJS_cr("  ajaxHandler.handleAjaxResponseFailed = function(responseContent) {");
  aWriter.addJS_cr("    var element = document.getElementById(targetDivId);");
  aWriter.addJS_cr("    element.innerHTML = \"<span class='" + UICSSClasses.TEXT.ERROR
   + "'>Ajax call failed</span>\";");
  aWriter.addJS_cr("  }");

  aWriter.addJS_cr("  ajaxHandler.sendRequest(url);");
  aWriter.addJS_cr("}");
 }

 /**
  * The Ajax callback that returns partial HTML.
  */
 private void ajaxCallback(
  UserServiceAjaxContext anAjaxContext,
  UserServiceAjaxResponse anAjaxResponse)
 {
  UserServiceWriter writer = anAjaxResponse.getWriter();
  writer.add("<p style=\"color:green\">Ajax callback succeeded!</p>");
  writer.add("<p>Current data and time is: ");

  DateFormat format = DateFormat.getDateTimeInstance(
   DateFormat.FULL,
   DateFormat.FULL,
   Locale.US);
  writer.addSafeInnerHTML(format.format(new Date()));

  writer.add("</p>");
 }
}
```

92.7 **REST data services**

A user service can access REST data services through HTTP requests.

The client should use the URL generated by: `UIResourceLocator.getURLForRest`[API]. This URL includes required information for the user authentication.

For more information on REST data services see the Built-in RESTful services [p 681].

The following sample implements a REST data service call whose response is printed in a `textarea`:

```
public class RestCallSampleService implements UserService<DataspaceEntitySelection>
{
 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  aConfigurator.setLeftButtons(aConfigurator.newCloseButton());
  aConfigurator.setContent(this::writePane);
 }

 private void writePane(UserServicePaneContext aPaneContext, UserServicePaneWriter aWriter)
 {
  // Generates the URL for REST data service call without additional parameters
  final String url = aWriter.getURLForRest("/ebx-dataservices/rest/{specificPath}", null);

  final String resultAreaId = "restResult";

  // Displays a link for REST data service call
  aWriter.add("<div ");
  aWriter.addSafeAttribute("class", UICSSClasses.CONTAINER_WITH_TEXT_PADDING);
  aWriter.add(">");
  aWriter.add("<p>This link will display the response after making a REST call</p>");
  aWriter.add("<a ");
  aWriter.addSafeAttribute("href",
   "javascript:sendRestRequest('" + url + "', '" + resultAreaId + "')");
  aWriter.add(">");
  aWriter.add("Make the call.");
  aWriter.add("</a>");
  aWriter.add("<textarea ");
  aWriter.addSafeAttribute("id", resultAreaId);
  aWriter.add(" readonly=\"readonly\" style=\"width: 100%;\" ></textarea>");
  aWriter.add("</div>");

  // JavaScript method that will send the HTTP REST request
  aWriter.addJS_cr("function sendRestRequest(url, targetId) {");
  aWriter.addJS_cr("  var xhttp = new XMLHttpRequest();");
  aWriter.addJS_cr("  xhttp.open('GET', url, true);");
  aWriter.addJS_cr("  xhttp.setRequestHeader('Content-type', 'application/json');");
  aWriter.addJS_cr("  xhttp.send();");
  aWriter.addJS_cr("  var element = document.getElementById(targetId);");
  aWriter.addJS_cr("  xhttp.onreadystatechange = function() {");
  aWriter.addJS_cr("   if (xhttp.readyState == 4)");
  aWriter.addJS_cr("      element.innerHTML = xhttp.responseText;");
  aWriter.addJS_cr("  }");
  aWriter.addJS_cr("}");
 }
}
```

92.8 **File upload**

A user service can display forms with file input fields.

The following sample displays a form with two input fields, a title and a file:

```
public class FileUploadService implements UserService<...>
{
 // This service defines a single object named "file".
 private final static ObjectKey _File_ObjectKey = ObjectKey.forName("file");

 // Paths for the "file" object.
 public final static Path _Title = Path.parse("title");
 public final static Path _File = Path.parse("file");

 ...

 @Override
 public void setupObjectContext(
  UserServiceSetupObjectContext<DataspaceEntitySelection> aContext,
  UserServiceObjectContextBuilder aBuilder)
 {
  if (aContext.isInitialDisplay())
  {
   // Create a definition for the "model" object.
```

```
   BeanDefinition def = aBuilder.createBeanDefinition();
   aBuilder.registerBean(_File_ObjectKey, def);

   BeanElement element;

   element = def.createElement(_Title, SchemaTypeName.XS_STRING);
   element.setLabel("Title");
   element.setMinOccurs(1);

   // Type for a file must be BeanDefinition.OSD_FILE_UPLOAD.
   element = def.createElement(_File, BeanDefinition.OSD_FILE_UPLOAD);
   element.setLabel("File");
   element.setMinOccurs(1);
  }
 }

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  aConfigurator.setTitle("File upload service");
  aConfigurator.setLeftButtons(aConfigurator.newSubmitButton("Upload", this::onUpload),
 aConfigurator.newCancelButton());

  // IMPORTANT: Following method must be called to enable file upload.
  // This will set form  encryption type to  "multipart/form-data".
  aConfigurator.setFileUploadEnabled(true);

  aConfigurator.setContent(this::writePane);
 }


 private void writePane(UserServicePaneContext aContext, UserServicePaneWriter aWriter)
 {
  final UIWidgetFileUploadFactory fileUploadFactory = new UIWidgetFileUploadFactory();

  aWriter.setCurrentObject(_File_ObjectKey);

  aWriter.startTableFormRow();

  // Title input.
  aWriter.addFormRow(_Title);

  // File upload input.
  UIWidgetFileUpload widget = aWriter.newCustomWidget(_File, fileUploadFactory);
  // Default filter for file names.
  widget.setAccept(".txt");
  aWriter.addFormRow(widget);

  aWriter.endTableFormRow();
 }

 private UserServiceEventOutcome onUpload(UserServiceEventContext anEventContext)
 {
  ValueContextForInputValidation valueContext = anEventContext.getValueContext(_File_ObjectKey);

  String title = (String) valueContext.getValue(_Title);
  UploadedFile file = (UploadedFile) valueContext.getValue(_File);

  InputStream in;
  try
  {
   in = file.getInputStream();
  }
  catch (IOException e)
  {
   // Should not happen.
   anEventContext.addError("Cannot read file.");
   return null;
  }

  // Do something with title and the input stream.
  return UserServiceNext.nextClose();
 }
}
```

For more information, see `UIWidgetFileUpload`[API].

92.9 File download

A user service can display URLs or buttons to download files. The actual downloading of a file is under the control of the user service.

The following sample displays a URL to download a file:

```
public class FileDownloadService implements UserService<DataspaceEntitySelection>
{
 ...

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<DataspaceEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  aConfigurator.setLeftButtons(aConfigurator.newCloseButton());
  aConfigurator.setContent(this::writePane);
 }

 private void writePane(UserServicePaneContext aContext, UserServicePaneWriter aWriter)
 {
  aWriter.add("<div ");
  aWriter.addSafeAttribute("class", UICSSClasses.CONTAINER_WITH_TEXT_PADDING);
  aWriter.add(">");

  // Generate and display the URL for the download.
  String downloadURL = aWriter.getURLForGetRequest(this::processDownloadRequest);

  aWriter.add("<a ");
  aWriter.addSafeAttribute("href", downloadURL);
  aWriter.add(">Click here to download a sample file</a>");

  aWriter.add("</div>");
 }

 private void processDownloadRequest(
  UserServiceGetContext aContext,
  UserServiceGetResponse aResponse)
 {
  // The file is plain text.
  aResponse.setContentType("text/plain;charset=UTF-8");
  // Remove the following statement to display the file directly in the browser.
  aResponse.setHeader("Content-Disposition", "attachment; filename=\"sample.txt\"");

  // Write a text file using UTF-8 encoding.
  PrintWriter out;
  try
  {
   out = new PrintWriter(new OutputStreamWriter(aResponse.getOutputStream(), "UTF-8"));
  }
  catch (IOException ex)
  {
   throw new RuntimeException(ex);
  }

  DateFormat format = DateFormat.getDateTimeInstance(
   DateFormat.FULL,
   DateFormat.MEDIUM,
   Locale.US);
  Date now = new Date();

  out.println("Hello !");
  out.println("This is a sample text file downloaded on " + format.format(now)
   + ", from EBX®.");

  out.close();
 }
}
```

92.10 User service without display

A user service may be designed to execute a task without display and return to the previous screen or redirect the user to another screen.

This type of service must implement the interface **UserServiceExtended** `UserServiceExtended`[API] and method `UserServiceExtended.initialize`[API].

The following sample deletes selected records in the current table view:

```
public class DeleteRecordsService implements UserServiceExtended<TableViewEntitySelection>
{
 ...

 @Override
 public UserServiceEventOutcome initialize(
  UserServiceInitializeContext<TableViewEntitySelection> aContext)
 {
  final List<AdaptationName> records = new ArrayList<>();

  // Deletes all selected rows in a single transaction.
  RequestResult requestResult = aContext.getEntitySelection().getSelectedRecords().execute();
  try
  {
   for (Adaptation record = requestResult.nextAdaptation(); record != null; record =
 requestResult.nextAdaptation())
   {
    records.add(record.getAdaptationName());
   }
  }
  finally
  {
   requestResult.close();
  }

  Procedure deleteProcedure = new Procedure()
  {
   @Override
   public void execute(ProcedureContext aContext) throws Exception
   {
    for (AdaptationName record : records)
    {
     aContext.doDelete(record, false);
    }
   }
  };

  UserServiceTransaction transaction = aContext.createTransaction();
  transaction.add(deleteProcedure);

  // Adds an information messages for current user.
  ProcedureResult procedureResult = transaction.execute(true);
  if (!procedureResult.hasFailed())
  {
   if (records.size() <= 1)
   {
    aContext.addInfo(records.size() + " record was deleted.");
   }
   else
   {
    aContext.addInfo(records.size() + " records were deleted.");
   }
  }

  // Do not display the user service and return to current view.
  return UserServiceNext.nextClose();
 }

 @Override
 public void setupObjectContext(
  UserServiceSetupObjectContext<TableViewEntitySelection> aContext,
  UserServiceObjectContextBuilder aBuilder)
 {
  //Do nothing.
 }

 @Override
 public void setupDisplay(
  UserServiceSetupDisplayContext<TableViewEntitySelection> aContext,
  UserServiceDisplayConfigurator aConfigurator)
 {
  //Do nothing.
 }

 @Override
 public void validate(UserServiceValidateContext<TableViewEntitySelection> aContext)
 {
  //Do nothing.
```

```
 }

 @Override
 public UserServiceEventOutcome processEventOutcome(
  UserServiceProcessEventOutcomeContext<TableViewEntitySelection> aContext,
  UserServiceEventOutcome anEventOutcome)
 {
  return anEventOutcome;
 }
}
```

Known limitation

If such service is called in the context of a Web component, an association, a perspective action or a hierarchy node, The service will be launched, initialized and closed, but the service's target entity will still be displayed.

CHAPITRE **93**

Declaring a user service

Ce chapitre contient les sections suivantes :

1. Declaration interface
2. Life cycle and threading model
3. Registration
4. Service properties
5. Service activation scope
6. Web component declaration
7. User service groups

93.1 Declaration interface

The following table lists, per nature, the interface that the declaration class of a user service must implement:

Nature	Declaration **Interface**
`Dataspace`	`UserServiceDeclaration.OnDataspace`
`Dataset`	`UserServiceDeclaration.OnDataset`
`TableView`	`UserServiceDeclaration.OnTableView`
`Record`	`UserServiceDeclaration.OnRecord`
`Hierarchy`	`UserServiceDeclaration.OnHierarchy`
`HierarchyNode`	`UserServiceDeclaration.OnHierarchyNode`
`Association`	`UserServiceDeclaration.OnAssociation`
`AssociationRecord`	`UserServiceDeclaration.OnAssociationRecord`

93.2 Life cycle and threading model

The user service declaration class is instantiated at the TIBCO EBX startup and must be coded to be thread-safe. This is usually not an issue as most implementations should be immutable classes.

93.3 Registration

A user service declaration must be registered by a module or a data model.

Registration by a module is achieved by the module registration servlet by a code similar to:

```
public class CustomRegistrationServlet extends ModuleRegistrationServlet
{
 @Override
 public void handleServiceRegistration(ModuleServiceRegistrationContext aContext)
 {
  // Register custom user service declaration.
  aContext.registerUserService(new CustomServiceDeclaration());
 }
}
```

For more information on the module registration servlet, see module registration [p 484] and `ModuleRegistrationServlet`[API].

Registration by a data model is achieved by a code similar to:

```
public class CustomSchemaExtensions implements SchemaExtensions
{
 @Override
 public void defineExtensions(SchemaExtensionsContext aContext)
 {
  // Register custom user service declaration.
  aContext.registerUserService(new CustomServiceDeclaration());
 }
}
```

For more information on data model extensions, see `SchemaExtensions`[API].

93.4 Service properties

The properties of a user service include its *label*, *description*, *confirmation message* and the *group* that owns the service. All are optional but it is a good practice to at least define the label.

For more information, see `UserServiceDeclaration.defineProperties`[API].

93.5 Service activation scope

The activation scope defines on which selection the service is available.

Example of a service activation definition:

```
public class CustomServiceDeclaration implements UserServiceDeclaration.OnTableView
{
 ...

 @Override
 public void defineActivation(ActivationContextOnTableView aContext)
 {
  // activates the service in all dataspaces except the "Reference" branch.
  aContext.includeAllDataspaces(DataspaceType.BRANCH);
  aContext.excludeDataspacesMatching(Repository.REFERENCE, DataspaceChildrenPolicy.NONE);

  // activates the service only on tables "table01" and "table03".
  aContext.includeSchemaNodesMatching(
   CustomDataModelPath._Root_Table01.getPathInSchema(),
   CustomDataModelPath._Root_Table03.getPathInSchema());

  // service will be enabled only when at least one record is selected.
```

```
  aContext.forbidEmptyRecordSelection();

  // service will not be displayed in hierarchical views (neither in the
  // top toolbar, nor in the hierarchy nodes' menu).
  aContext.setDisplayForLocations(
   ActionDisplaySpec.HIDDEN,
   ToolbarLocation.HIERARCHICAL_VIEW_TOP,
   ToolbarLocation.HIERARCHICAL_VIEW_NODE);

  // service will be considered as disabled if not explicitly enabled
  // via the UI.
  aContext.setDefaultPermission(UserServicePermission.getDisabled());
 }
}
```

For more information about declaring the activation scope, see `UserServiceDeclaration.defineActivation`[API].

For more information about the resolution of the user service availability, see Rsolution de permissions pour les services [p 311].

93.6 Web component declaration

Parameters declaration and availability in workflows and perspectives

User services are automatically available as web components with a set of built-in parameters depending on the service's nature. To define custom parameters and/or set the service web component as available when configuring a workflow user task, a perspective menu action or a toolbar web component action, `UserServiceDeclaration.declareWebComponent`[API] must be used.

Example of a web component declaration:

```
public class CustomServiceDeclaration implements UserServiceDeclaration.OnDataset
{
 ...

 @Override
 public void declareWebComponent(WebComponentDeclarationContext aContext)
 {
  // makes this web component available when configuring a workflow user task.
  aContext.setAvailableAsWorkflowUserTask(true);

  // adds a custom input parameter.
  aContext.addInputParameter(
   "source",
   UserMessage.createInfo("Source"),
   UserMessage.createInfo("Source of the imported data."));

  // modifies the built-in "instance" parameter to be "input/output" instead of "input".
  aContext.getBuiltInParameterForOverride("instance").setOutput(true);
 }
}
```

See Utiliser TIBCO EBX comme composant web [p 211] for more information.

User service extension

It is possible to extend existing user services (built-in or custom) in order to add input/output parameters when using these services as web components.

In order to do so, a user service extension must first be registered by a module or a data model.

Registration by a module is achieved by the module registration servlet by code similar to:

```
public class CustomRegistrationServlet extends ModuleRegistrationServlet
{
 ...

 @Override
 public void handleServiceRegistration(ModuleServiceRegistrationContext aContext)
```

```
 {
  // Register user service extension declaration.
  aContext.registerUserServiceExtension(new ServiceExtensionDeclaration());
 }
}
```

For more information on the module registration servlet, see module registration [p 484] and `ModuleRegistrationServlet`[API].

Registration by a data model is achieved by a code similar to:

```
public class CustomSchemaExtensions implements SchemaExtensions
{
 ...

 @Override
 public void defineExtensions(SchemaExtensionsContext aContext)
 {
  // Register user service extension declaration.
  aContext.registerUserServiceExtension(new ServiceExtensionDeclaration());
 }
}
```

For more information on the data model extension, see `SchemaExtensions`[API].

93.7 User service groups

User service groups are used to organize the display of user services in menus and permission management screens.

The following types of service groups are available:

- Built-in User Service Groups [p 612] provided by EBX,
- Custom User Service Groups [p 613] declared in a module.

The link between groups and services is made upon service declaration. See Associating a service to a group [p 613].

Built-in User Service Groups

Available built-in service groups:

Service Group Key	Description
`@ebx-importExport`	Group containing all built-in import and export services provided by EBX. In the default menus, these services are displayed in an "Import / Export" sub-menu.
`@ebx-views`	Group containing services to display in the 'Views' menu. Unlike other service groups, services associated with this group are not displayed in the default menus, but only in the 'Views' menu displayed in the non-customizable part of the table top toolbar. These services can still be added manually to a custom toolbar.

Declaring a User Service Group

User Service Groups must be declared while registering the module, using the method `ModuleServiceRegistrationContext.registerServiceGroup`[API]:

```
public class CustomRegistrationServlet extends ModuleRegistrationServlet
{
 ...

 @Override
 public void handleServiceRegistration(ModuleServiceRegistrationContext aContext)
 {
  // In CustomModuleConstants,
  // CUSTOM_SERVICE_GROUP_KEY = ServiceGroupKey.forServiceGroupInModule("customModule", "customGroup")

  // registers CUSTOM_SERVICE_GROUP_KEY service group
  aContext.registerServiceGroup(
   CustomModuleConstants.CUSTOM_SERVICE_GROUP_KEY,
   UserMessage.createInfo("Custom group"),
   UserMessage.createInfo("This group contains services related to..."));
 }
}
```

Associating a service to a group

The association of a service with a group is made at its **declaration** `UserServiceDeclaration`[API], using the method `UserServicePropertiesDefinitionContext.setGroup`[API]:

```
public class CustomServiceDeclaration implements UserServiceDeclaration.OnDataset
{
 ...

 @Override
 public void defineProperties(UserServicePropertiesDefinitionContext aContext)
 {
  // associates the current service to the CUSTOM_SERVICE_GROUP_KEY group
  aContext.setGroup(CustomModuleConstants.CUSTOM_SERVICE_GROUP_KEY);
 }
}
```

A service can be associated with either a built-in or a custom service group. In the latter case, this service will be displayed in this built-in group, just like other built-in services belonging to this group.

CHAPITRE **94**

Development recommendations

Ce chapitre contient les sections suivantes :

1. HTML
2. CSS
3. JavaScript

94.1 HTML

It is recommended to minimize the inclusion of specific HTML styles and tags to allow the default styles of TIBCO EBX to apply to custom interfaces. The approach of the API is to automatically apply a standardized style to all elements on HTML pages, while simplifying the implementation process for the developer.

XHTML

EBX is a Rich Internet Application developed in XHTML 1.0 Transitional. It means that the structure of the HTML is strict XML file and that all tags must be closed, including "br" tags. This structure allows for greater control over CSS rules, with fewer differences in browser rendering.

iFrames

Using iFrame is allowed in EBX, especially in collaboration with a URL of a `UIHttpManagerComponent`[API]. For technical reasons, it is advised to set the `src` attribute of an iFrame using JavaScript only. In this way, the iFrame will be loaded once the page is fully rendered and when all the built-in HTML components are ready.

Example

The following example, developed from any `UIComponentWriter`[API], uses a `UIHttpManagerComponent`[API] to build the URL of an iFrame, and set it in the right way:

```
// using iFrame in the current page requires a sub session component
UIHttpManagerComponent managerComponent = writer.createWebComponentForSubSession();

// [...] managerComponent configuration

String iFrameURL = managerComponent.getURIWithParameters();

String iFrameId = "mySweetIFrame";

// place the iFrame in the page, with an empty src attribute
writer.add("<iframe id=\"").add(iFrameId).add("\" src=\"\" >").add("</iframe>");

// launch the iFrame from JavaScript
```

```
writer.addJS("document.getElementById(\"").addJS(iFrameId).addJS("\").src = \"").addJS(iFrameURL).addJS("\";");
```

94.2 CSS

Public CSS classes

The constant catalog `UICSSClasses`[API] offers the main CSS classes used in the software to style the components. These CSS classes ensure a proper long-term integration into the software, because they follow the background colors, borders, customizable text in the administration; the floating margins and paddings fluctuate according to the variable density; to the style of the icons, etc.

Voir aussi *`UICSSUtils`*[API]

Advanced CSS

EBX allows to integrate to all its pages one or more external Cascading Style Sheet. These external CSS, considered as resources, need to be declared in the Module registration [p 484].

In order to ensure the proper functioning of your CSS rules and properties without altering the software, the following recommendations should be respected. Failure to respect these rules could lead to:

- Improper functioning of the software, both aesthetically and functionally: risk of losing the display of some of the data and some input components may stop working.
- Improper functioning of your CSS rules and properties, since the native CSS rules will impact the CSS implementation.

Reserved prefixes for CSS identifiers and class names

The following prefixes should not be used to create CSS #ids and .classes.

ebx_	Internal built-in
yui	Yahoo User Interface global
ygtv	Yahoo User Interface tree view
layout-doc	Yahoo User Interface layout
cke_	CK editor (used by HTML editor widget)
fa	Font Awesome (icons used by perspectives and toolbars)

CSS classes used internally by EBX

The following CSS classes should never be included in a ruleset that has no contextual selector.

If you do not prefix your CSS selector using one of the CSS classes below, it will cause conflicts and corrupt the UI of EBX.

selected	YUI selected tree node
hd	YUI floating pane header
bd	YUI floating pane body
ft	YUI floating pane footer
container-close	YUI inner popup close button
underlay	YUI inner popup shadow
hastitle	YUI menu group with title
topscrollbar	YUI menu top scroll zone
bottomscrollbar	YUI menu bottom scroll zone
withtitle	YUI calendar
link-close	YUI calendar close button
collapse	YUI layout closed pane indicator
pull-right	Font Awesome parameter
pull-left	Font Awesome parameter

Examples to avoid conflicts

Don't

```
.selected {
 background-color: red;
}
```

Do

```
#myCustomComponent li.selected {
 background-color: red;
}
```

94.3 JavaScript

Public JS functions

The catalog of JavaScript functions `JavaScriptCatalog`[API] offers a list of functions to use directly (through copy-paste) in the JS files.

JavaScript call during page generation in Java

When generating the HTML of a Java component, it is possible to add specific JavaScript code with the API `UIJavaScriptWriter`[API].

This JavaScript is executed once the whole page is loaded. It is possible to instantly manage the HTML elements written with `UIBodyWriter.add`[API]. Setting on-load functions (such as `window.onload = myFunctionToCallOnload;`) is not supported because the execution context comes after the on-load event.

Advanced JavaScript

EBX allows to include one or more external JavaScript files. These external JavaScript files, considered as resources, need to be declared in the Module registration [p 484]. For performance reasons, it is recommended to include the JavaScript resource only when necessary (in a User service or a specific form, for example). The API `UIDependencyRegisterer`[API] allows a developer to specify the conditions for which the JavaScript resources will be integrated into a given page according to its context.

In order to ensure the proper functioning of your JavaScript resources without altering the software, the following recommendations should be respected. Failure to respect them could lead to:

- Improper functioning of the software: if functions or global variables of the software were to be erased, some input or display components (including the whole screen) may stop working.
- Improper functioning of your JavaScript instructions, since global variables or function names could be erased.

Reserved JS prefixes

The following prefixes are reserved and should not be used to create variables, functions, methods, classes, etc.

ebx_	Internal built-in API
EBX_	Internal built-in API
YAHOO	Yahoo User Interface API

SOAP data services

CHAPITRE **95**

Introduction

Ce chapitre contient les sections suivantes :

1. Overview
2. Activation and configuration
3. Interactions
4. Security
5. Monitoring
6. SOAP and REST comparative
7. Limitations

95.1 Overview

Data services allow external systems to interact with the data governed in the TIBCO EBX repository using the SOAP/Web Services Description Language (WSDL) standards.

In order to invoke SOAP operations [p 639], for an integration use case, a WSDL [p 631] must be generated from a data model. It will be possible to perform operations such as:

- Selecting, inserting, updating, deleting, or counting records
- Selecting dataset values
- Getting the differences on a table between dataspaces or snapshots, or between two datasets based on the same data model
- Getting the credentials of records

Other generic WSDLs can be generated and allow performing operations such as:

- Creating, merging, or closing a dataspace
- Creating or closing a snapshot
- Validating a dataset, dataspace, or a snapshot
- Starting, resuming or ending a data workflow

- Administrative operations to manage access to the UI or to system information

> **Note**
>
> See SOAP and REST comparative [p 628].

95.2 Activation and configuration

Data services are enabled by deploying the `ebx-dataservices` web application along with the other EBX modules. See Java EE deployment overview [p 341] for more information.

In case of specific deployment, for example using reverse-proxy mode, see URLs computing [p 380] for more information.

The default method for accessing data services is over HTTP, although it is also possible to use JMS for the SOAP operations. See JMS configuration [p 379] and Using JMS [p 622] for more information.

95.3 Interactions

Input and output message encoding

All input messages must be *exclusively* in UTF-8. All output messages are in UTF-8.

Tracking information

Depending on the data services operation being called, it may be possible to specify session tracking information.

- Example for a SOAP operation, the request header contains:

```
<SOAP-ENV:Header>
 <!-- optional security header here -->
 <m:session xmlns:m="urn:ebx-schemas:dataservices_1.0">
  <trackingInformation>String</trackingInformation>
 </m:session>
</SOAP-ENV:Header>
```

For more information, see `Session.getTrackingInfo`[API] in the Java API.

Session parameters

Depending on the data services operation being called, it is possible to specify session input parameters. They are defined in the request body.

Input parameters are available on custom Java components with a session object, such as: triggers, access rules, `custom` web services. They are also available on data workflow operations.

- Example for a SOAP operation, the optional request header contains:

```
<SOAP-ENV:Header>
 <!-- optional security header here -->
 <m:session xmlns:m="urn:ebx-schemas:dataservices_1.0">
  <!-- optional trackingInformation header here -->
  <inputParameters>
   <parameter>
    <name>String</name>
    <value>String</value>
   </parameter>
   <!-- for some other parameters, copy complex
        element 'parameter' -->
  </inputParameters>
 </m:session>
</SOAP-ENV:Header>
```

For more information, see `Session.getInputParameterValue`[API] in the Java API.

Exception handling

In case of unexpected server error upon execution of:

- A SOAP operation, a SOAP exception response is returned to the caller via the `soap:Fault` element. For example:

```
<soapenv:Fault>
 <faultcode>soapenv:java.lang.IllegalArgumentException</faultcode>
 <faultstring />
 <faultactor>admin</faultactor>
 <detail>
  <m:StandardException xmlns:m="urn:ebx-schemas:dataservices_1.0">
   <code>java.lang.IllegalArgumentException</code>
   <label/>
   <description>java.lang.IllegalArgumentException:
    Parent home not found at
    com.orchestranetworks.XX.YY.ZZ.AA.BB(AA.java:44) at
    com.orchestranetworks.XX.YY.ZZ.CC.DD(CC.java:40) ...
   </description>
  </m:StandardException>
 </detail>
</soapenv:Fault>
```

Using JMS

It is possible to access SOAP operations using JMS instead of HTTP. The JMS architecture relies on one JMS request queue (mandatory), on one JMS failure queue (optional), and on JMS response queues, see configuration JMS [p 379]. The mandatory queue is the input queue. Request messages must be put in the input queue, and response messages are put by EBX in the `replyTo` queue of the JMS request. The optional queue is the failure queue which allows you to replay an input message if necessary. If the queue is set and activated in the configuration file and an exception occurs while handling a request message, this input message will be copied in the failure queue.

The relationship between a request and a response is made by copying the `messageId` message identifier field of the JMS request into the `correlId` correlation identifier field of the response.

JMS location points must be defined in the Lineage administration in order to specialize the generated WSDL. If no specific location point is given, the default value will be `jms:queue:jms/EBX_QueueIn`.

95.4 Security

Authentication

Authentication is mandatory to access to data. Several authentication methods are available and described below. The descriptions are ordered by priority (EBX applies the highest priority authentication method first).

- 'Basic Authentication Scheme' method is based on the HTTP-Header `Authorization` in base 64 encoding, as described in RFC 2617 (Basic Authentication Scheme).

```
If the user agent wishes to send the userid "Alibaba" and password "open sesame",
it will use the following header field:
> Authorization: Basic QWxpYmFiYTpvcGVuIHNlc2FtZQ==
```

- 'Standard Authentication Scheme' is based on the HTTP `Request`. User and password are extracted from request parameters. For more information on request parameters, see Parameters [p 634] section.

For more information on this authentication scheme, see `Directory.authenticateUserFromLoginPassword`[API].

- The 'SOAP Security Header Authentication Scheme' method is based on the Web Services Security UsernameToken Profile 1.0 specification.

  By default, the type `PasswordText` is supported. This is done with the following SOAP-Header defined in the WSDL:

```
<SOAP-ENV:Header>
 <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
  <wsse:UsernameToken>
   <wsse:Username>String</wsse:Username>
   <wsse:Password Type="wsse:PasswordText">String</wsse:Password>
  </wsse:UsernameToken>
 </wsse:Security>
</SOAP-ENV:Header>
```

  > **Note**
  >
  > Only available for SOAP operations [p 639].

- 'Specific authentication Scheme' is based on the HTTP `Request`. An implementation of this method can, for example, extract a password-digest or a ticket from the HTTP request. See `Directory.authenticateUserFromHttpRequest`[API] for more information.
- The 'SOAP Specific Header Authentication Scheme'.

  For more information, see Overriding the SOAP security header [p 623].

Global permissions

Global access permissions can be independently defined for the SOAP and WSDL connector accesses. For more information see Global permissions [p 407].

Overriding the SOAP security header

It is possible to override the default WSS header in order to define another security authentication mechanism. Such an override is taken into account for both HTTP and JMS. To define and override,

use the 'SOAP Header Security declaration' configuration settings under Administration > Lineage, which includes the following fields:

Schema location	The URI of the Security XML Schema to import into the WSDL.
Target namespace	The target namespace of elements in the schema.
Namespace prefix	The prefix for the target namespace.
Message name	The message name to use in the WSDL.
Root element name	The root element name of the security header. The name must be the same as the one declared in the schema.
wsdl:part element name	The name of the `wsdl:part` of the message.

The purpose of overriding the default security header is to change the declaration of the WSDL message matching the security header so that it contains the following:

```
<wsdl:definitions ... xmlns:MyPrefix="MyTargetNameSpace" ...
  ...
  <xs:schema ...>
    <xs:import namespace="MyTargetNameSpace" schemaLocation="MySchemaURI"/>
    ...
  </xs:schema>
  ...
  <wsdl:message name="MySecurityMessage">
    <wsdl:part name="MyPartElementName" element="MyPrefix:MySecurityRootElement"/>
  </wsdl:message>
  ...
  <wsdl:operation name="...">
    <soap:operation soapAction="..." style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
      <soap:header message="impl:MySecurityMessage" part="MyPartElementName" use="literal"/>
  ...
  </wsdl:operation>
</wsdl:definitions>
```

The corresponding XML Schema header declaration would be as follows:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="MyNameSpace"
  xmlns:MyPrefix="MyNameSpace">
  <element name="MySecurityRootElement" type="MyPrefix:SpecificSecurity"/>
  <complexType name="SpecificSecurity">
    <sequence>
      <element name="AuthToken" type="string"/>
    </sequence>
  </complexType>
</schema>
```

A SOAP message using the XML schema and configuration above would have the following header:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Header>
    <m:MySecurityRootElement xmlns:m="MyNameSpace">
      <AuthToken>String</AuthToken>
    </m:MySecurityRootElement>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ...
  </SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

To handle this non-default header, you must implement the method: `Directory.authenticateUserFromSOAPHeader`[API].

> **Note**
>
> Only available for SOAP operations [p 639].

Lookup mechanism

Because EBX offers several authentication methods, a lookup mechanism based on conditions was set to know which method should be applied for a given request. The method application conditions are evaluated according to the authentication scheme priority. If the conditions are not satisfied, the server evaluates the next method. The following table presents the available authentication methods for each

supported protocol and their application conditions. They are ordered from the highest priority to the lowest.

Operation / Protocol	Authentication methods and application conditions
SOAP / JMS	SOAP Security Header [p 623] • The SOAP request is received over the JMS protocol. • The SOAP header content must contains a `Security` element. SOAP Specific Header [p 623] • The SOAP request is received over the JMS protocol. • The SOAP header content must not contain a `Security` element.
SOAP / HTTP	Basic [p 622] • The HTTP request must hold an `Authorization` header. • `Authorization` header value must start with the word `Basic`. • No `login` is provided in the URL parameters. Standard [p 622] • The HTTP request must not hold an `Authorization` header. • A `login` and a `password` are provided in the URL parameters. SOAP Security Header [p 623] • The SOAP header content must contain a `Security` element. • The HTTP request must not hold an `Authorization` header. • No `login` is provided in the URL parameters. Specific [p 623] • The HTTP request must not satisfy the conditions of the previous authentication methods. SOAP Specific Header [p 623] • The SOAP header content must not contain a `Security` element. • The HTTP request must not hold an `Authorization` header. • No `login` is provided in the URL parameters.
WSDL / HTTP	Basic [p 622] • The HTTP request must not hold an `Authorization` header. • `Authorization` header value must start with the word `Basic`. • No `login` is provided in the URL parameters. Standard [p 622] • The HTTP request must not hold an `Authorization` header. • A `login` and a `password` are provided in the URL parameters. Specific [p 623] • The HTTP request must not satisfy the conditions of the previous authentication methods.

In case of multiple authentication methods present in the same request, EBX will return an HTTP code `401 Unauthorized`.

95.5 Monitoring

Data service events can be monitored through the log category `ebx.dataServices`, as declared in the EBX main configuration file. For example, `ebx.log4j.category.log.dataServices= INFO, ebxFile:dataservices`.

Voir aussi

Configuring the EBX logs *[p 375]*

TIBCO EBX main configuration file *[p 369]*

95.6 **SOAP and REST comparative**

Operations	**SOAP**	**REST**
Data		
Select or count records (with filter and/or view publication)	X	X
Selector for possible enumeration values (with filter)		X
Insert, update or delete records	X	X
Select or count history records (with filter and/or view publication)		X
Select node values from dataset	X	X
Update node value from dataset		X
Get table or dataset changes between dataspaces or snapshots	X	
Refresh a replication unit	X	
Get credentials for records	X	
Dataspaces		
Create, close, merge a dataspace	X	
Create, close a snapshot	X	
Validate a dataspace or a snapshot	X	
Validate a dataset	X	
Locking a dataspace	X	
Workflow		
Start, resume or end a workflow	X	
Administration		
Manage the default directory content 'Users', 'Roles'... tables.	X	X
Open, close the user interface	X	X

Operations	SOAP	REST
Select, insert, update, delete operations for administration dataset		X
Select the system information	X	X
Other		
Develop web services from the Java API		X (*)

(*) See REST Toolkit [p 745] for more information.

95.7 Limitations

Date, time & dateTime format

Data services only support the following date and time formats:

Type	Format	Example
xs:date	yyyy-MM-dd	2007-12-31
xs:time	HH:mm:ss or HH:mm:ss.SSS	11:55:00
xs:dateTime	yyyy-MM-ddTHH:mm:ss or yyyy-MM-ddTHH:mm:ss.SSS	2007-12-31T11:55:00

SOAP naming convention

Due to the naming convention of the data service operations, each table defined within a data model must have a unique name for the WSDL generation.

CHAPITRE **96**

WSDL generation

Ce chapitre contient les sections suivantes :

1. Supported standard
2. Operation types
3. WSDL download methods
4. HTTP examples

96.1 Supported standard

TIBCO EBX generates a WSDL that complies with the W3C Web Services Description Language 1.1 standard.

96.2 Operation types

A WSDL can be generated for different types of operations:

Operation type	WSDL description
`custom`	WSDL for EBX add-ons.
`dataset`	WSDL for dataset and replication operations.
`directory`	WSDL for default EBX directory operations. It is also possible to filter data using the `tablePaths` [p 635] or `operations` [p 635] parameters.
`repository`	WSDL for dataspace or snapshot management operations.
`tables`	WSDL for operations on the tables of a specific dataset.
`userInterface`	Deprecated since version 5.8.1. This operation type has been replaced by `administration`. While the user interface management operations are still available for backward compatibility reasons, it is recommended to no longer use this type. WSDL for user interface management operations (these operations can only be accessed by administrators).
`administration`	WSDL for administration operations like: • user interface management • system information retrieval These operations can only be accessed by administrators.
`workflow`	WSDL for EBX workflow management operations.

96.3 WSDL download methods

EBX supports the following methods:

- from the user interface
- from HTTP protocol

A WSDL can only be downloaded by authorized profiles:

Operation type	Access right permissions
`custom`	All profiles, if at least one web service is registered.
`dataset`	All profiles.
`directory`	All profiles, if the following conditions are valid: • No specific directory implementation is used. (The built-in Administrator role is only subject to this condition). • Global access permissions are defined for the administration. • 'Directory' dataset permissions have writing access for the current profile.
`repository`	All profiles.
`tables`	All profiles.
`userInterface`	Deprecated since version 5.8.1. This operation type has been replaced by `administration`. While the user interface management operations are still available for backward compatibility reasons, it is recommended to no longer use this type. Built-in administrator role or delegated administrator profiles, if all conditions are valid: • Global access permissions are defined for the administration. • 'User interface' dataset permissions have writing access for the current profile.
`administration`	Built-in administrator role or delegated administrator profiles, if all conditions are valid: • Global access permissions are defined for the administration. • 'Administration' dataset permissions have write access for the current profile.
`workflow`	All profiles.

WSDL download from the user interface

An authorized user can download an EBX WSDL from the data services administration area.

> **Note**
>
> See Gnrer un WSDL pour accder un espace de donnes [p 202] in the user guide for more information.

WSDL download from HTTP protocol

An application can download an EBX WSDL using `GET` or `POST` HTTP method. The application has to be authenticated using a profile with appropriate rights.

URL format

```
http[s]://<host>[:<port>]/<ebx-dataservices>/{type}[/{dataspace}[/{dataset}]]?
{queryParameters}
```

Where:

- `<ebx-dataservices>` corresponds to the 'ebx-dataservices.war' web application's path. The path is composed by multiple, or none, URI segments followed by the web application's name.
- `{type}` corresponds to the operation type [p 632].
- `{dataspace}` corresponds to the dataspace or snapshot identifier.
- `{dataset}` corresponds to the dataset name.
- `{queryParameters}` corresponds to common or dedicated operation parameters passed through the URL.

Parameters

A request parameter can be specified by one of the following methods:

- a `PathParam` which corresponds to a path segment from the URL (recommended)

- a `QueryParam` which corresponds to a standard HTTP parameter with value.

Parameter name	`PathParam`	`QueryParam`	Required	Description
`WSDL`	no	yes	yes	Specifies the WSDL download operation. Empty value.
`login`	no	yes	no	Specifies the user identifier. Required when the standard authentication method is used. `String` type value.
`password`	no	yes	no	Specifies the user password. Required when the standard authentication method is used. `String` type value.
`type`	yes	no	yes	Specifies the operation type [p 632]. Possible values are: `custom`, `dataset`, `directory`, `administration`, `userInterface`, `repository`, `tables` or `workflow`. `String` type value.
`branch` `version`	yes	yes	(*)	Specifies the dataspace or the snapshot identifier. (*) required for `tables` and `dataset` types, ignored otherwise. `String` type value.
`instance`	yes	yes	(*)	Specifies the dataset name. `String` type value.
`tablePaths`	no	yes	no	Specifies the list of table paths. Optional for `tables` or `directory` types, ignored otherwise. If not defined, all tables are selected. Each table path is separated by a comma character. `String` type value.
`operations`	no	yes	no	Allows generating a WSDL for an operations subset. Optional for `tables` or `directory` operation types, ignored otherwise. If not defined, all operations for the given type are generated. This parameter's value is a concatenation of one or more of the following characters: • C = Count record(s) • D = Delete record(s) • E = Get credentials • G = Get changes • I = Insert record(s) • U = Update record(s)

Parameter name	PathParam	QueryParam	Required	Description
				• R = Read operations (equivalent to CEGS) • S = Select record(s) • W = Write operations (equivalent to DIU) String type value.
namespaceURI	yes	yes	(**)	Specifies the unique name space URI of the custom web service. (**)Is required when type parameter is set to custom, ignored otherwise. URI type value.
attachmentFilename	no	yes	(***)	Specifies the attachment file name. (***) optional if isContentInAttachment parameter is defined and set to true, ignored otherwise. String type value.
isContentInAttachment	no	yes	no	Specifies if the WSDL is downloaded as an attachment. Boolean type value. Default value is false.
targetNamespace	no	yes	no	Overrides the target namespace URI of the WSDL. URI type value. Default value is urn:ebx:ebx-dataservices.

Message body

No message body is required.

HTTP codes

An HTTP code is always returned. Errors are indicated by a code above 400.

Status code	Information
`200` *(OK)*	The WSDL content was successfully generated and is returned by the request (optionally in an attachment [p 636]).
`400` *(Bad request)*	The request is incorrect. This occurs when: • A request element is incorrect. • The unicity check on table names contains at least one error. **Note** See WSDL and table operations [p 570] for more information.
`401` *(Unauthorized)*	Request requires an authenticated user.
`403` *(Forbidden)*	Request is not allowed for the authenticated user.
`405` *(Method not allowed)*	Request is not allowed in this configuration.
`500` *(Internal error)*	Request generates an error (a stack trace and a detailed error message are returned).

Response body

The response body depends on the returned status code and on the requested WSDL content.

- `200` *(OK)*: the HTTP header `Content-Type` is set to `text/xml;charset=UTF-8`.

  If the content is in attachment, the HTTP header `Content-Disposition` is set to `attachment; filename*=UTF-8''<filename.wsdl>`.
- `4xx`: A detailed message is returned in the body. The HTTP header `Content-Type` is set to `text/html;charset=utf-8`.

96.4 HTTP examples

Some of the following examples are displayed in two methods: `PathParam` and `QueryParam`.

- The WSDL will contain all repository operations, using standard authentication method.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/repository?
  WSDL&login=<login>&password=<password>
  ```
- The WSDL will contain all workflow operations.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/workflow?WSDL
  ```
- The WSDL will contain all tables operations for the `'dataset1'` dataset in `'dataspace1'` dataspace.

  `PathParam`

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables/dataspace1/dataset1?WSDL
```

```
QueryParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables?
WSDL&branch=<dataspace1>&instance=<dataset1>
```

- The WSDL will contain all tables with only read operations for the `'dataset1'` dataset in `'dataspace1'` dataspace.

```
PathParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables/dataspace1/dataset1?
WSDL&operations=R
```

```
QueryParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables?
WSDL&branch=dataspace1&instance=dataset1&operations=R
```

- The WSDL will contain the two selected tables operations for the `'dataset1'` dataset in `'dataspace1'` dataspace.

```
PathParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables/dataspace1/dataset1?
WSDL&tablePaths=/root/table1,/root/table2
```

```
QueryParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/tables?
WSDL&branch=dataspace1&instance=dataset1&tablePaths=/root/table1,/root/table2
```

- The WSDL will contain custom web service operations for the dedicated URI.

```
PathParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/custom/urn:ebx-
test:com.orchestranetworks.dataservices.WSDemo?WSDL
```

```
QueryParam
```

```
http[s]://<host>[:<port>]/<ebx-dataservices>/custom?WSDL&namespaceURI=urn:ebx-
test:com.orchestranetworks.dataservices.WSDemo
```

CHAPITRE 97

SOAP operations

Ce chapitre contient les sections suivantes :

1. Operations generated from a data model
2. Operations on datasets and dataspaces
3. Operations on data workflows
4. Administrative services

97.1 Operations generated from a data model

For a data model used in an TIBCO EBX repository, it is possible to dynamically generate a corresponding WSDL, that defines its operations. When using this WSDL, it will be possible to read and/or write in the EBX repository. For example, for a table located at the path `/root/XX/exampleTable`, the generated requests would follow the structure of its underlying data model and include the name of the table `<m:{operation}_exampleTable xmlns:m="urn:ebx-schemas:dataservices_1.0">`.

Attention

Since the WSDL and the SOAP operations tightly depend on the data model structure, it is important to redistribute the up-to-date WSDL after any data model change.

Content policy

Access to the content of records, the presence or absence of XML elements, depend on the resolved permissions [p 299] of the authenticated user session. Additional aspects, detailed below, can impact the content.

Disabling fields from data model

The `hiddenInDataServices` property, defined in the data model, allows always hiding fields in data services, regardless of the user profile. This parameter has an impact on the generated WSDL: any hidden field or group will be absent from the request and response structure.

Modifying the `hiddenInDataServices` parameter value has the following impact on a client which would still use the former WSDL:

- On request, if the data model property has been changed to `true`, and if the concerned field is present in the WSDL request, an exception will be thrown.

- On response, if the schema property has been changed to `false`, WSDL validation will return an error if it is activated.

This setting of "Default view" is defined inside data model.

Voir aussi

Hiding a field in Data Services *[p 564]*

Permissions *[p 299]*

Association field

Read-access on table records can export the association fields as displayed in UI Manager. This feature can be coupled with the 'hiddenInDataServices' model parameter.

> **Note**
>
> Limitations: change and update operations do not manage association fields. Also, the select operation only exports the first level of association elements (the content of associated objects cannot contain association elements).

Common request parameters

Several parameters are common to several operations and are detailed below.

Element	Description	Required
branch	The identifier of the dataspace to which the dataset belongs.	Either this parameter or the 'version' parameter must be defined. Required for the 'insert', 'update' and 'delete' operations.
version	The identifier of the snapshot to which the dataset belongs.	Either this parameter or the 'branch' parameter must be defined
instance	The unique name of the dataset which contains the table to query.	Yes
predicate	XPath predicate [p 249] defines the records on which the request is applied. If empty, all records will be retrieved, except for the 'delete' operation where this field is mandatory.	Only required for the 'delete' operation
data	Contains the records to be modified, represented within the structure of their data model. The whole operation is equivalent to an XML import. The details of the operations performed on data content are specified in the section Import [p 235].	Only required for the insert and update operations
viewPublication	This parameter can be combined with the predicate [p 641] parameter as a logical `AND` operation. The behavior of this parameter is described in the section EBX as a Web Component [p 215]. It cannot be used if the 'viewId' parameter is used, and cannot be used on hierarchical views.	No
viewId	*Deprecated since version 5.2.3*. This parameter has been replaced by the parameter 'viewPublication'. While it remains available for backward compatibility, it will eventually be removed in a future version. This parameter cannot be used if the 'viewPublication' parameter is used.	No
blockingConstraintsDisabled	This property is available for all table updates data service operations. If `true`, the validation process disables blocking constraints defined in the data model. If this parameter is not present, the default is `false`. See Blocking and non-blocking constraints [p 546] for more information.	No
details	The `details` element specifies the following option: The optional attribute `locale` (default 'en-US') defines the language of the blockingConstraintsDisabled [p 641] parameter in which the validation messages must be returned.	No

Element	Description	Required
disableRedirectionToLastBroadcast	This property is available for all data service operations. If `true`, access to a delivery dataspace on a D3 primary node is not redirected to the last broadcast snapshot. Otherwise, access to such a dataspace is always redirected to the last snapshot broadcast. If this parameter is not present, the default is `false` (redirection on a D3 master enabled), unless the configuration property ebx.dataservices.disableRedirectionToLastBroadcast.default [p 378] has been set. If the specified dataspace is not a delivery dataspace on a D3 primary node, this parameter is ignored.	No

Select operations

Select request on table

```
<m:select_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
 <viewPublication>String</viewPublication>
 <exportCredentials>boolean</exportCredentials>
 <pagination>
  <previousPageLastRecordPredicate>String</previousPageLastRecordPredicate>
  <pageSize>Integer</pageSize>
 </pagination>
</m:select_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
version	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
predicate	See the description under Common parameters [p 641]. This parameter can be combined with the viewPublication [p 641] parameter as a logical `AND` operation.	
viewPublication	See the description under Common parameters [p 641].	
includesTechnicalData	The response will contain technical data if `true`. See also the optimistic locking [p 658] section. Each returned record will contain additional attributes for this technical information, for instance: `...<table ebxd:lastTime="2010-06-28T10:10:31.046" ebxd:lastUser="Uadmin" ebxd:uuid="9E7D0530-828C-11DF-B733-0012D01B6E76">... .`	No
exportCredentials	If `true` the select will also return the credentials for each record.	No
pagination	Enables pagination, see child elements below.	No
pageSize (nested under the `pagination` element)	When pagination is enabled, defines the number of records to retrieve.	When pagination is enabled, yes
previousPageLastRecordPredicate (nested under the `pagination` element)	When pagination is enabled, XPath predicate that defines the record after which the page must fetched, this value is provided by the previous response, as the element `lastRecordPredicate`. If the passed record is not found, the first page will be returned.	No
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	

Select response on table

```
<ns1:select_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <data>
  <XX>
   <TableName>
    <a>key1</a>
    <b>valueb</b>
    <c>1</c>
    <d>1</d>
   </TableName>
  </XX>
 </data>
 <credentials>
  <XX>
   <TableName predicate="./a='key1'">
    <a>W</a>
    <b>W</b>
```

```
    <c>W</c>
    <d>W</d>
   </TableName>
  </XX>
 </credentials>
 <lastRecordPredicate>./a='key1'</lastRecordPredicate>
</ns1:select_{TableName}Response>
```

with:

Element	Description
data	Content of records that are displayed following the table path.
credentials	Contains the access right for each node of each record.
lastRecordPredicate	Only returned if the pagination is enabled, this defines the last records in order to be used on the next request in the element `previousPageLastRecordPredicated`.

See also the optimistic locking [p 658] section.

Select request on dataset

This operation returns dataset content without table.

```
<m:selectInstance xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
</m:selectInstance>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
version	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	

Select response on dataset

```
<ns1:selectInstanceResponse xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <data>
  <settings>
   <XX>
    <a>key1</a>
    <b>valueb</b>
    <c>1</c>
    <d>true</d>
   </XX>
  </settings>
 </data>
</ns1:selectInstanceResponse>
```

with:

Element	Description
data	Dataset content without table.

Delete operation

Deletes records or, for a child dataset, defines the record state as "occulting" or "inherited" according to the record context. Records are selected by the `predicate` parameter.

Delete request

```
<m:delete_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
 <includeOcculting>boolean</includeOcculting>
 <inheritIfInOccultingMode>boolean</inheritIfInOccultingMode>
 <checkNotChangedSinceLastTime>dateTime</checkNotChangedSinceLastTime>
 <blockingConstraintsDisabled>boolean</blockingConstraintsDisabled>
 <details locale="Locale"/>
</m:delete_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
predicate	See the description under Common parameters [p 641].	
includeOcculting	Includes the records in occulting mode. Default value is `false`.	No
inheritIfInOccultingMode	Inherits the record from its parent if it is in occulting mode. Default value is `false`.	No
occultIfInherit	*Deprecated since version 5.7.0* Occults the record if it is in inherit mode. Default value is `false`.	No
checkNotChangedSinceLastTime	Timestamp used to ensure that the record has not been modified since the last read. Also see the optimistic locking [p 658] section.	No
blockingConstraintsDisabled	See the description under Common parameters [p 641].	
details	See the description under Common parameters [p 641].	
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	

Delete response

If one of the provided parameters is illegal, if a required parameter is missing, if the action is not authorized or if no record is selected, an exception is returned. Otherwise, the specific response is returned:

```
<ns1:delete_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <blockingConstraintMessage>String</blockingConstraintMessage>
</ns1:delete_{TableName}Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully. '95' indicates that at least one operation has violated a blocking constraint, resulting in the overall operation being aborted.
blockingConstraintMessage	This element is present if the status is equal to '95' with a localized message. The locale of the message is retrieved from the request parameter or from the user session.

Count operation

Count request

```
<m:count_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
</m:count_{TableName}>
```

with:

Element	Description
branch	See the description under Common parameters [p 641].
version	See the description under Common parameters [p 641].
instance	See the description under Common parameters [p 641].
predicate	See the description under Common parameters [p 641].
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].

Count response

```
<ns1:count_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <count>Integer</count>
</ns1:count_{TableName}Response>
```

with:

Element	Description
count	The number of records that correspond to the predicate in the request.

Update operation

Update request

```
<m:update_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <updateOrInsert>boolean</updateOrInsert>
 <byDelta>boolean</byDelta>
 <blockingConstraintsDisabled>boolean</blockingConstraintsDisabled>
 <details locale="Locale"/>
 <data>
  <XX>
   <TableName>
    <a>String</a>
    <b>String</b>
    <c>String</c>
    <d>String</d>
    ...
   </TableName>
  </XX>
 </data>
</m:update_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
updateOrInsert	If `true` and the record does not currently exist, the operation creates the record. `boolean` type, the default value is `false`.	No
byDelta	If `true` and an element does not currently exist in the incoming message, the target value is not changed. If `false` and node is declared `hiddenInDataServices`, the target value is not changed. The complete behavior is described in the sections Oprations d'insertion et de mise jour [p 236].	No
blockingConstraintsDisabled	See the description under Common parameters [p 641].	
details	See the description under Common parameters [p 641].	
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	
data	See the description under Common parameters [p 641].	

Voir aussi*Optimistic locking [p 658]*

Update response

```
<ns1:update_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <blockingConstraintMessage>String</blockingConstraintMessage>
</ns1:update_{TableName}Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully. '95' indicates that at least one operation has violated a blocking constraint, resulting in the overall operation being aborted.
blockingConstraintMessage	This element is present if the status is equal to '95' with a localized message. The locale of the message is retrieved from the request parameter or from the user session.

Insert operation

Insert request

```
<m:insert_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <byDelta>boolean</byDelta>
 <blockingConstraintsDisabled>boolean</blockingConstraintsDisabled>
 <details locale="Locale"/>
 <data>
  <XX>
   <TableName>
    <a>String</a>
    <b>String</b>
    <c>String</c>
    <d>String</d>
    ...
   </TableName>
  </XX>
 </data>
</m:insert_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
byDelta	If `true` and an element does not currently exist in the incoming message, the target value is not changed. If `false` and node is declared `hiddenInDataServices`, the target value is not changed. The complete behavior is described in the sections Oprations d'insertion et de mise_jour [p 236].	No
blockingConstraintsDisabled	See the description under Common parameters [p 641].	
details	See the description under Common parameters [p 641].	
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	
data	See the description under Common parameters [p 641].	

Insert response

```
<ns1:insert_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <blockingConstraintMessage>String</blockingConstraintMessage>
 <inserted>
  <predicate>./a='String'</predicate>
 </inserted>
</ns1:insert_{TableName}Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully. '95' indicates that at least one operation has violated a blocking constraint, resulting in the overall operation being aborted.
blockingConstraintMessage	This element is present if the status is equal to '95' with a localized message. The locale of the message is retrieved from the request parameter or from the user session.
predicate	A predicate matching the primary key of the inserted record. When several records are inserted, the predicates follow the declaration order of the records in the input message.

Get changes operations

Returns changes according to the Content policy [p 639].

Get changes requests

Changes between two datasets:

```
<m:getChangesOnDataSet_{schemaName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <compareWithBranch>String</compareWithBranch>
 <compareWithVersion>String</compareWithVersion>
 <compareWithInstance>String</compareWithInstance>
 <resolvedMode>boolean</resolvedMode>
 <includeInstanceUpdates>boolean</includeInstanceUpdates>
</m:getChangesOnDataSet_{schemaName}>
```

Changes between two tables:

```
<m:getChanges_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <compareWithBranch>String</compareWithBranch>
 <compareWithVersion>String</compareWithVersion>
 <resolvedMode>boolean</resolvedMode>
</m:getChanges_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
version	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
compareWithBranch	The identifier of the dataspace with which to compare.	One of either this parameter or the 'compareWithVersion [p 651]' parameter must be defined.
compareWithVersion	The identifier of the snapshot with which to compare.	One of either this parameter or the 'compareWithBranch [p 651]' parameter must be defined.
compareWithInstance	The identifier of the dataset with which to compare. If it is undefined, instance [p 651] parameter is used.	No
resolvedMode	Defines whether or not the difference is calculated in resolved mode. Default is `true`. See **Resolved mode** `DifferenceHelper.resolvedMode`[API] for more information.	No
includeInstanceUpdates	Defines if the content updates of the dataset are included. Default is `false`.	No
pagination	Enables pagination context for the operations `getChanges` and `getChangesOnDataSet`. Allows client to define pagination context size. Each page contains a collection of inserted, updated and/or deleted records of tables according to the maximum size. Get changes persisted context is built at first call according to the page size parameter in request. The pagination context is persisted on the server file system [p 404] and allows invoking the next page until last page or when a timeout is reached. For creation: Defines `pageSize` parameter. For next: Defines `context` element with `identifier` from previous response. Enables pagination, see child elements below.	No
pageSize (nested under `pagination` element)	Defines maximum number of records in each page. Minimal size is `50`.	No (Only for creation)
context (nested under `pagination` element)	Defines content of pagination context.	No (Only for next)

Element	Description	Required
identifier (nested under `context` element)	Pagination context identifier.	Yes
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	

> **Note**
>
> If none of the *compareWithBranch* or *compareWithVersion* parameters are specified, the comparison will be made with their parent:
>
> - if the current dataspace or snapshot is a dataspace, the comparison is made with its initial snapshot (includes all changes made in the dataspace);
> - if the current dataspace or snapshot is a snapshot, the comparison is made with its parent dataspace (includes all changes made in the parent dataspace since the current snapshot was created);
> - returns an exception if the current dataspace is the 'Reference' dataspace.

Voir aussi *`DifferenceHelper`*[API]

Get changes responses

Changes between two datasets:

```
<ns1:getChangesOnDataSet_{schemaName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <updated>
  <changes>
   <path>... Path of changed terminal value ...</path>
   <path>...</path>
  </changes>
  <data>
   ... see the whole content of dataset values (without table) ...
  </data>
 </updated>
 <getChanges_{TableName1}>
  ... see the getChanges between tables response example ...
 </getChanges_{TableName1}>
 <getChanges_{TableName2}>
  ... see the getChanges between tables response example ...
 </getChanges_{TableName2}>
 ...
</ns1:getChangesOnDataSet_{schemaName}Response>
```

Changes between two tables:

```
<ns1:getChanges_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <inserted>
  <XX>
   <TableName>
    <a>AVALUE3</a>
    <b>BVALUE3</b>
    <c>CVALUE3</c>
    <d>DVALUE3</d>
   </TableName>
  </XX>
 </inserted>
 <updated>
  <changes>
   <change predicate="./a='AVALUE2'">
    <path>/b</path>
    <path>/c</path>
   </change>
  </changes>
  <data>
   <XX>
    <TableName>
     <a>AVALUE2</a>
```

```
      <b>BVALUE2.1</b>
      <c>CVALUE2.1</c>
      <d>DVALUE2</d>
     </TableName>
    </XX>
   </data>
  </updated>
  <deleted>
   <predicate>./a='AVALUE1'</predicate>
  </deleted>
</ns1:getChanges_{TableName}Response>
```

with:

Element	Description	Required
inserted	Contains inserted record(s) from choice `compareWithBranch` or `compareWithVersion`. Content under this element corresponding to an XML export of inserted records.	No
updated	Contains updated record(s) or dataset content.	No
changes (nested under `updated` element)	Only the group of field have been updated.	Yes
change (nested under `changes` element)	Group of fields have been updated with own `XPath predicate` attribute of the record.	Yes
path (nested under `change` element)	Path in the record.	Yes
path (nested under `changes` element)	Path in the dataset.	Yes
data (nested under `updated` element)	Content under this element corresponding to an XML export of dataset or updated records.	No
deleted	Records have been deleted from context of request. Content corresponding to a list of `predicate` element who contains the XPath predicate of record.	No
pagination	When pagination is enabled on request. Get changes persisted context allows invoking the next page until last page or when the context timeout is reached. Contains a next page: Defines `context` element with `identifier`. Is the last page: Defines `context` element without `identifier`. Enables pagination, see child elements below.	No
context (nested under `pagination` element)	Defines content of pagination context.	Yes (Only for next and last)
identifier (nested under `context` element)	Pagination context identifier. Not defined at last returned page.	No
pageNumber (nested under `context` element)	Current page number in pagination context.	Yes
totalPages (nested under `context` element)	Total pages in pagination context.	Yes

Get changes operation with pagination enabled

Only `pagination` element and sub elements have been described.

For creation:

Extract of request:

```
...
 <pagination>
  <!-- on first request for creation -->
  <pageSize>Integer</pageSize>
 </pagination>
...
```

Extract of response:

```
...
 <pagination>
  <!-- on next request to continue -->
  <context>
   <identifier>String</identifier>
   <pageNumber>Integer</pageNumber>
   <totalPages>Integer</totalPages>
  </context>
 </pagination>
...
```

For next:

Extract of request:

```
...
 <pagination>
  <context>
   <identifier>String</identifier>
  </context>
 </pagination>
...
```

Extract of response:

```
...
 <pagination>
  <!-- on next request to continue -->
  <context>
   <identifier>String</identifier>
   <pageNumber>Integer</pageNumber>
   <totalPages>Integer</totalPages>
  </context>
 </pagination>
...
```

For last:

Extract of request:

```
...
 <pagination>
  <context>
   <identifier>String</identifier>
  </context>
 </pagination>
...
```

Extract of response:

```
...
 <pagination>
  <context>
   <pageNumber>Integer</pageNumber>
   <totalPages>Integer</totalPages>
  </context>
 </pagination>
...
```

Get credentials operation

Get credentials request

```
<m:getCredentials_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
 <viewPublication>String</viewPublication>
</m:getCredentials_{TableName}>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
version	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
predicate	See the description under Common parameters [p 641].	
viewPublication	See the description under Common parameters [p 641].	

Get credentials response

```
<ns1:getCredentials_{TableName}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <XX>
  <TableName>
   <a>R</a>
   <b>W</b>
   <c>H</c>
   <d>W</d>
   ...
  </TableName>
 </XX>
</ns1:getCredentials_{TableName}Response>
```

With the following possible values:

- R: for read-only
- W: for read-write
- H: for hidden

Multiple chained operations

Multiple operations request

It is possible to run multiple operations across tables in the dataset, while ensuring a consistent response. The operations are executed sequentially, according to the order defined on the client side.

All operations are executed in a single transaction with a `SERIALIZABLE` isolation level. If all requests in the multiple operation are read-only, they are allowed to run fully concurrently along with other read-only transactions, even in the same dataspace.

When an error occurs during one operation in the sequence, all updates are rolled back and the client receives a `StandardException` error message with details.

See Concurrency and isolation levels [p 490].

```
<m:multi_ xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <blockingConstraintsDisabled>boolean</blockingConstraintsDisabled>
 <details locale="Locale"/>
 <request id="id1">
  <{operation}_{TableName}>
  ...
  </{operation}_{TableName}>
 </request>
 <request id="id2">
  <{operation}_{TableName}>
  ...
  </{operation}_{TableName}>
 </request>
</m:multi_>
```

with:

Element	Description	Required
branch	See the description under Common parameters [p 641].	
version	See the description under Common parameters [p 641].	
instance	See the description under Common parameters [p 641].	
blockingConstraintsDisabled	See the description under Common parameters [p 641].	
details	See the description under Common parameters [p 641].	
disableRedirectionToLastBroadcast	See the description under Common parameters [p 642].	
request	This element contains one operation, like a single operation without branch, version and instance parameters. This element can be repeated multiple times for additional operations. Each request can be identified by an 'id' attribute. In a response, this 'id' attribute is returned for identification purposes. Operations such as `count`, `select`, `getChanges`, `getCredentials`, `insert`, `delete` or `update`.	Yes

Note:

- Does not accept a limit on the number of `request` elements.
- The request `id` attribute must be unique in multi-operation requests.
- If all operations are read only (`count`, `select`, `getChanges`, or `getCredentials`) then the whole transaction is set as read-only for performance considerations.

Limitation:

- The `multi` operation applies to one model and one dataset (parameter `instance`).
- The `select` operation cannot use the pagination parameter.

Voir aussi

Procedure[API]

Repository[API]

Multiple operations response

See each response operation for details.

```
<ns1:multi_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <response id="id1">
  <{operation}_{TableName}Response>
  ...
  </{operation}_{TableName}Response>
 </response>
 <response id="id2">
  <{operation}_{TableName}Response>
  ...
  </{operation}_{TableName}Response>
 </response>
</ns1:multi_Response>
```

with:

Element	Description
response	This element contains the response of one operation. It is be repeated multiple times for additional operations. Each response is identified by an 'id' attribute set in the request or automatically generated. The content of the element corresponds to the response of a single operation, such as `count`, `select`, `getChanges`, `getCredentials`, `insert`, `delete` or `update`.

Optimistic locking

To prevent an update or a delete operation on a record that was read earlier but may have changed in the meantime, an optimistic locking mechanism is provided.

A select request can include technical information by adding the element `includesTechnicalData`:

```
<m:select_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
 <includesTechnicalData>boolean</includesTechnicalData>
</m:select_{TableName}>
```

The value of the `lastTime` attribute can then be used in the following update request. If the record has been changed since the specified time, the update will be cancelled. The attribute `lastTime` has to be added on the record to prevent the update of a modified record.

```
<m:update_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
    <version>String</version>
 <instance>String</instance>
 <updateOrInsert>true</updateOrInsert>
 <data>
  <XX>
   <TableName ebxd:lastTime="2010-06-28T10:10:31.046">
    <a>String</a>
    <b>String</b>
    <c>String</c>
    <d>String</d>
    ...
   </TableName>
  </XX>
 </data>
```

```
</m:update_{TableName}>
```

The value of the `lastTime` attribute can also be used to prevent deletion on a modified record:

```
<m:delete_{TableName} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <predicate>String</predicate>
 <checkNotChangedSinceLastTime>2010-06-28T10:10:31.046</checkNotChangedSinceLastTime>
</m:delete_{TableName}>
```

> **Note**
>
> The element `checkNotChangedSinceLastTime` may be used more than once but only for the same record. This implies that if the `predicate` element returns more than one record, the request will fail.

97.2 **Operations on datasets and dataspaces**

Parameters for operations on dataspaces and snapshots are as follows:

Element	*Description*	*Required*
branch	Identifier of the target dataspace on which the operation is applied. When not specified, the 'Reference' dataspace is used except for the merge dataspace operation where it is required.	One of either this parameter or the 'version' parameter must be defined. Required for the dataspace merge, locking, unlocking and replication refresh operations.
version	Identifier of the target snapshot on which the operation is applied.	One of either this parameter or the 'branch' parameter must be defined
versionName	Identifier of the snapshot to create. If empty, it will be defined on the server side.	No
childBranchName	Identifier of the dataspace child to create. If empty, it will be defined on the server side.	No
instance	The unique name of the dataset on which the operation is applied.	Required for the replication refresh operation.
ensureActivation	Defines if validation must also check whether this instance is activated.	Yes
details	Defines if validation returns details. The optional attribute `severityThreshold` defines the lowest severity level of message to return. The possible values are, in descending order of severity, 'fatal', 'error', 'warning', or 'info'. For example, setting the value to 'error' will return error and fatal validation messages. If this attribute is not defined, all levels of messages are returned by default. The optional attribute `locale` (default 'en-US') defines the language in which the validation messages are to be returned.	No. If not specified, no details are returned.
owner	Defines the owner. Must respect the inner format as returned by `Profile.format`[API].	No
branchToCopyPermissionFrom	Defines the identifier of the dataspace from which to copy the permissions.	No
documentation	Documentation for a dedicated language.	No

Element	*Description*	*Required*
	Multiple `documentation` elements may be used for several languages.	
locale (nested under the `documentation` element)	Locale of the documentation.	Only required when the `documentation` element is used
label (nested under the `documentation` element)	Label for the language.	No
description (nested under the `documentation` element)	Description for the language.	No

Validate a dataspace

Validate dataspace request

```
<m:validate xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
</m:validate>
```

Validate dataspace response

```
<ns1:validate_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <validationReport>
  <instanceName>String</instanceName>
  <fatals>boolean</fatals>
  <errors>boolean</errors>
  <infos>boolean</infos>
  <warnings>boolean</warnings>
 </validationReport>
</ns1:validate_Response>
```

Validate a dataset

Validate dataset request

```
<m:validateInstance xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <version>String</version>
 <instance>String</instance>
 <ensureActivation>boolean</ensureActivation>
 <details severityThreshold="fatal|error|warning|info" locale="Locale"/>
</m:validateInstance>
```

Validate dataset response

```
<ns1:validateInstance_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <validationReport>
  <instanceName>String</instanceName>
  <fatals>boolean</fatals>
  <errors>boolean</errors>
  <infos>boolean</infos>
  <warnings>boolean</warnings>
  <details>
   <reportItem>
    <severity>{fatal|error|warning|info}</severity>
    <message>
     <internalId />
     <text>String</text>
    </message>
    <subject>
     <table>Path</table>
     <predicate>String</predicate>
```

```
     <path>Path</path>
    </subject>
   </reportItem>
  </details>
 </validationReport>
</ns1:validateInstance_Response>
```

Create a dataspace

Create dataspace request

```
<m:createBranch xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <owner>String</owner>
 <branchToCopyPermissionFrom>String</branchToCopyPermissionFrom>
 <documentation>
  <locale>Locale</locale>
  <label>String</label>
  <description>String</description>
 </documentation>
 <childBranchName>String</childBranchName>
</m:createBranch>
```

Create dataspace response

```
<ns1:createBranch_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <childBranchName>String</childBranchName>
</ns1:createBranch_Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully.

Create a snapshot

Create snapshot request

```
<m:createVersion xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <versionName>String</versionName>
 <owner>String</owner>
 <documentation>
  <locale>Locale</locale>
  <label>String</label>
  <description>String</description>
 </documentation>
</m:createVersion>
```

Create snapshot response

```
<ns1:createVersion_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <versionName>String</versionName>
</ns1:createVersion_Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully.

Locking a dataspace

Lock dataspace request

```
<m:lockBranch xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <durationToWaitForLock>Integer</durationToWaitForLock>
 <message>
  <locale>Locale</locale>
  <label>String</label>
 </message>
</m:lockBranch>
```

with:

Element	*Description*	*Required*
durationToWaitForLock	This parameter defines the maximum duration (in seconds) that the operation waits for a lock before aborting.	No, does not wait by default
message	User message of the lock. Multiple `message` elements may be used.	No
locale (nested under the `message` element)	Locale of the user message.	Only required when the `message` element is used
label (nested under the `message` element)	The user message.	No

Lock dataspace response

```
<ns1:lockBranch_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:lockBranch_Response>
```

with:

Element	**Description**
status	'00' indicates that the operation has been executed successfully. '94' indicates that the dataspace has been already locked by another user. Otherwise, a SOAP exception is thrown.

Unlocking a dataspace

Unlock dataspace request

```
<m:unlockBranch xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
</m:unlockBranch>
```

Unlock dataspace response

```
<ns1:unlockBranch_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:unlockBranch_Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully. Otherwise, a SOAP exception is thrown.

Merge a dataspace

Merge dataspace request

```
<m:mergeBranch xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <deleteDataOnMerge>boolean</deleteDataOnMerge>
 <deleteHistoryOnMerge>boolean</deleteHistoryOnMerge>
</m:mergeBranch>
```

with:

Element	*Description*	*Required*
deleteDataOnMerge	This parameter is available for the merge dataspace operation. Sets whether the specified dataspace and its associated snapshots will be deleted upon merge. When this parameter is not specified in the request, the default value is `false`. It is possible to redefine the default value by specifying the property `ebx.dataservices.dataDeletionOnCloseOrMerge.default` in the EBX main configuration file [p 378]. See Deleting data and history [p 402] for more information.	No
deleteHistoryOnMerge	This parameter is available for the merge dataspace operation. Sets whether the history associated with the specified dataspace will be deleted upon merge. Default value is `false`. When this parameter is not specified in the request, the default value is `false`. It is possible to redefine the default value by specifying the property `ebx.dataservices.historyDeletionOnCloseOrMerge.default` in the EBX main configuration file [p 378]. See Deleting data and history [p 402] for more information.	No

> **Note**
>
> The merge decision step is bypassed during merges performed through data services. In such cases, the data in the child dataspace automatically overrides the data in the parent dataspace.

Merge dataspace response

```
<ns1:mergeBranch_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:mergeBranch_Response>
```

with:

Element	Description
status	'00' indicates that the operation has been executed successfully.

Close a dataspace or snapshot

Close dataspace or snapshot request

Close dataspace request:

```
<m:closeBranch xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <deleteDataOnClose>boolean</deleteDataOnClose>
 <deleteHistoryOnClose>boolean</deleteHistoryOnClose>
</m:closeBranch>
```

Close snapshot request:

```
<m:closeVersion xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <version>String</version>
 <deleteDataOnClose>boolean</deleteDataOnClose>
</m:closeVersion>
```

with:

Element	*Description*	*Required*
deleteDataOnClose	This parameter is available for the close dataspace and close snapshot operations. Sets whether the specified snapshot, or dataspace and its associated snapshots, will be deleted upon closure. When this parameter is not specified in the request, the default value is `false`. It is possible to redefine this default value by specifying the property `ebx.dataservices.dataDeletionOnCloseOrMerge.default` in the EBX main configuration file [p 378]. See Deleting data and history [p 402] for more information.	No
deleteHistoryOnClose	This parameter is available for the close dataspace operation. Sets whether the history associated with the specified dataspace will be deleted upon closure. Default value is `false`. When this parameter is not specified in the request, the default value is `false`. It is possible to redefine the default value by specifying the property `ebx.dataservices.historyDeletionOnCloseOrMerge.default` in the EBX main configuration file [p 378]. See Deleting data and history [p 402] for more information.	No

Close dataspace or snapshot response

Close dataspace response:

```
<ns1:closeBranch_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:closeBranch_Response>
```

Close snapshot request:

```
<ns1:closeVersion_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:closeVersion_Response>
```

Replication refresh

Replication refresh request

```
<m:replicationRefresh_${schema} xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>String</branch>
 <instance>String</instance>
 <unitName>String</unitName>
</m:replicationRefresh_${schema}>
```

with:

Element	*Description*	*Required*
branch	See the description under Common parameters [p 660].	Yes
instance	See the description under Common parameters [p 660].	Yes
unitName	Name of the replication unit. **Voir aussi***Replication refresh information [p 282]*	Yes

Replication refresh response

```
<ns1:replicationRefresh_${schema}Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:replicationRefresh_${schema}Response>
```

with:

Element	**Description**
status	'00' indicates that the operation has been executed successfully.

97.3 Operations on data workflows

Parameters for data workflows operations are retrieved from the SOAP header in the session.

Deprecated since version 5.7.0 to define parameters in the SOAP message body.

See session parameters [p 621] for more information.

Element	Description	Required
parameters	*Deprecated since version 5.7.0* While it remains available for backward compatibility, it will eventually be removed in a future major version. **Note** The `parameters` element is ignored if at least one session parameter has been defined.	No
parameter (nested under the `parameters` element). Multiple `parameter` elements may be used.	An input parameter for the workflow.	No
name (nested under the `parameter` element)	Name of the parameter.	Yes
value (nested under the `parameter` element)	Value of the parameter.	No

Start a workflow

Start a workflow from a workflow launcher. It is possible to start a workflow with localized documentation and specific input parameters (with name and optional value).

> **Note**
>
> The workflow creator is initialized from the session and the workflow priority is retrieved from the last published version.

Sample request:

```
<m:workflowProcessInstanceStart xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <publishedProcessKey>String</publishedProcessKey>
 <documentation>
  <locale>Locale</locale>
  <label>String</label>
  <description>String</description>
 </documentation>
</m:workflowProcessInstanceStart>
```

with:

Element	Description	Required
publishedProcessKey	Identifier of the workflow launcher.	Yes
documentation	See the description under Common parameters [p 660].	No
parameters	*Deprecated since version 5.7.0* See the description under Common parameters [p 667].	No

Sample response:

```
<m:workflowProcessInstanceStart_Response xmlns:m="urn:ebx-schemas:dataservices_1.0">
```

```
 <processInstanceKey>String</processInstanceKey>
</m:workflowProcessInstanceStart_Response>
```

with:

Element	Description	Required
processInstanceId	*Deprecated since version 5.6.1* This parameter has been replaced by the 'processInstanceKey' parameter. While it remains available for backward compatibility, it will eventually be removed in a future major version.	No
processInstanceKey	Workflow identifier.	No

Resume a workflow

Resume a workflow in a wait step from a resume identifier. It is possible to define specific input parameters (with name and optional value).

Sample request:

```
<m:workflowProcessInstanceResume xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <resumeId>String</resumeId>
</m:workflowProcessInstanceResume>
```

with:

Element	Description	Required
resumeId	Resume identifier of the waiting task.	Yes
parameters	*Deprecated since version 5.7.0* See the description under Common parameters [p 667].	No

Sample response:

```
<m:workflowProcessInstanceResume_Response xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
 <processInstanceKey>String</processInstanceKey>
</m:workflowProcessInstanceResume_Response>
```

with:

Element	Description	Required
status	'00' indicates that the operation has been executed successfully. '20' indicates that the workflow has not been found. '21' indicates that the event has already been received.	Yes
processInstanceKey	Identifier of the workflow. This parameter is returned if the operation has been executed successfully.	No

End a workflow

End a workflow from its identifier.

Sample request:

```
<m:workflowProcessInstanceEnd xmlns:m="urn:ebx-schemas:dataservices_1.0">
```

```
 <processInstanceKey>String</processInstanceKey>
</m:workflowProcessInstanceEnd>
```

with:

Element	Description	Required
processInstanceKey	Identifier of the workflow.	Either this parameter or 'publishedProcessKey' and 'processInstanceId' parameters must be defined.
publishedProcessKey	*Deprecated since version 5.6.1* Due to a limitation this parameter has been replaced by the 'processInstanceKey' parameter. While it remains available for backward compatibility, it will eventually be removed in a future major version.	No
processInstanceId	*Deprecated since version 5.6.1* Due to a limitation this parameter has been replaced by the 'processInstanceKey' parameter. While it remains available for backward compatibility, it will eventually be removed in a future major version.	No

Sample response:

```
<m:workflowProcessInstanceEnd_Response xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</m:workflowProcessInstanceEnd_Response>
```

with:

Element	Description	Required
status	'00' indicates that the operation has been executed successfully.	Yes

97.4 Administrative services

Directory services

The services on directory provide operations on the 'Users' and 'Roles' tables of the default directory. To execute an operation related to these services, the authenticated user must be a member of the built-in role 'Administrator'.

The technical dataspace and dataset must be set to `ebx-directory`. For all SOAP operation syntaxes, see Operations generated from a data model [p 639] for more information.

Create a user in the directory

This example of a SOAP insert request adds a user to the EBX directory.

```
<m:insert_user xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <branch>ebx-directory</branch>
 <instance>ebx-directory</instance>
 <data>
 <directory>
  <users>
   <login>login</login>
   <lastName>lastname</lastName>
```

```
   <firstName>firstname</firstName>
   <email>firstname.lastname@email.com</email>
   <password>***</password>
   <passwordMustChange>true</passwordMustChange>
   <builtInRoles>
    <administrator>false</administrator>
    <readOnly>false</readOnly>
   </builtInRoles>
   <comments>a comment</comments>
  </users>
 </directory>
 </data>
</m:insert_user>
```

For the insert SOAP response syntax, see insert response [p 649] for more information.

User interface operations

See Application locking [p 412] for more information.

Parameters for operations on the user interface are as follows:

Element	Description	Required
closedMessage	Message to be displayed to users when the user interface is closed to access.	No

Close user interface request

The close operation removes all user sessions that are not acceptable in this mode.

```
<m:close xmlns:m="urn:ebx-schemas:dataservices_1.0">
 <closedMessage>Access is temporarily forbidden.</closedMessage>
</m:close>
```

Close user interface response

```
<ns1:close_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:close_Response>
```

Open user interface request

```
<m:open xmlns:m="urn:ebx-schemas:dataservices_1.0"/>
```

Open user interface response

```
<ns1:open_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
 <status>String</status>
</ns1:open_Response>
```

System information operation

This operation returns the EBX system information. The information returned is the same as the information contained in the log header `kernel.log` or in the UI tab 'Administration' > 'System Information'. The response contains several keys, labels, and values representing the configuration and status of EBX. To execute this operation, the authenticated user must be a member of the built-in role 'Administrator'.

Parameters

The following parameter is applicable.

Parameter	Description	Required
`details`	Defines attributes that must be applied to response messages. The attribute `locale` (default: EBX default `locale`) defines the language in which the system item messages must be returned.	No, but if specified, the `locale` attribute must be provided.

System information request

This SOAP request will return all EBX instance's system information and format them using "`en_US`" `locale`.

```
<m:systemInformation xmlns:m="urn:ebx-schemas:dataservices_1.0">
    <details locale="en_US" />
</m:systemInformation>
```

System information response

```
<ns1:systemInformation_Response xmlns:ns1="urn:ebx-schemas:dataservices_1.0">
    <bootInfoEBX>
        <label>String</label>
        <infoItem>
            <key>String</key>
            <label>String</label>
            <content>String</content>
            <content>String</content>
            ...
        </infoItem>
        ...
    </bootInfoEBX>
    <repositoryInfo>
        <label>String</label>
        <infoItem>
            <key>String</key>
            <label>String</label>
            <content>String</content>
            <content>String</content>
            ...
        </infoItem>
        ...
    </repositoryInfo>
    <bootInfoVM>
        <label>String</label>
        <infoItem>
            <key>String</key>
            <label>String</label>
            <content>String</content>
            ...
        </infoItem>
        ...
    </bootInfoVM>
</ns1:systemInformation_Response>
```

REST data services

CHAPITRE 98

Introduction

Ce chapitre contient les sections suivantes :

1. Overview
2. Activation and configuration
3. Interactions
4. Security
5. Monitoring
6. SOAP and REST comparative
7. Limitations

98.1 Overview

REST data services allow external systems to interact with data governed in the TIBCO EBX repository using the RESTful built-in services.

The request and response syntax for built-in services are described in the chapter Built-in RESTful services [p 681].

Built-in REST data services allow to perform operations such as:

- Selecting, inserting, updating, deleting, or counting records
- Selecting or counting history records
- Selecting, updating, or counting dataset values
- Administrative operations to manage access to the UI or to system information

> **Note**
>
> See SOAP and REST comparative [p 679].

98.2 Activation and configuration

REST and SOAP Data services are activated by deploying the `ebx-dataservices` web application along with the other EBX modules. See Java EE deployment overview [p 341] for more information.

In case of specific deployment, for example using reverse-proxy mode, see URLs computing [p 380] for more information.

Currently only protocol HTTP(S) is supported.

98.3 Interactions

Input and output message encoding

All input and output messages must be *exclusively* in UTF-8 for REST built-in.

Tracking information

Depending on the data services operation being called, it may be possible to specify session tracking information.

- Example for a RESTful operation, the JSON request contains:

```
{
 "procedureContext": // JSON Object (optional)
 {
    "trackingInformation": "String" // JSON String (optional)
 }, ...
}
```

For more information, see `Session.getTrackingInfo`[API] in the Java API.

Session parameters

Depending on the data services operation being called, it is possible to specify session input parameters. They are defined in the request body.

Input parameters are available on custom Java components with a session object, such as: triggers, access rules, `custom` web services. They are also available on data workflow operations.

- Example for a RESTful operation, the JSON request contains:

```
{
 "procedureContext": // JSON Object (optional)
 {
    "trackingInformation": "String", // JSON String (optional)
    "inputParameters": // JSON Array (optional)
    [
     // JSON Object for each parameter
     {
      "name" : "String" // JSON String (required)
      "value" : "String" // JSON String (optional)
     },
     ...
    ]
 }, ...
}
```

For more information, see `Session.getInputParameterValue`[API] in the Java API.

Exception handling

When an error occurs, a JSON exception response is returned to the caller. For example:

```
{
  "code": 999, // JSON Number, HTTP status code
  "errors": [
   {
    "severity": "...",     // JSON String, severity (optional)
    "rowIndex": 999,       // JSON Number, request row index (optional)
    "userCode": "...",     // JSON String, user code (optional)
    "message": "...",      // JSON String, message
    "details": "...",      // JSON String, URL (optional)
    "pathInRecord": "...", // JSON String, Path in record (optional)
    "pathInDataset": "..." // JSON String, Path in dataset (optional)
   }
  ]
```

```
}
```

The response contains an HTTP status code and a table of errors. The severity of each error is specified by a character, with one of the possible values (`F`: fatal, `E`: error, `W`: warning, `I`: information).

The HTTP error `422` *(Unprocessable entity)* corresponds to a functional error. It contains a user code under the `userCode` key and is a JSON `String` type.

Voir aussi*HTTP codes* *[p 687]*

Voir aussi*`Severity.toParsableString`*[API]

98.4 Security

Authentication

Authentication is mandatory to access built-in services. Several authentication methods are available and described below. The descriptions are ordered by priority (EBX applies the highest priority authentication method first).

- 'Token Authentication Scheme' method is based on the HTTP-Header `Authorization`, as described in RFC 2617.

  ```
  > Authorization: <tokenType> <accessToken>
  ```

  For more information on this authentication scheme, see Token authentication operations [p 690].

  Voir aussi*HTTP Authorization header policy* *[p 380]*

- 'Basic Authentication Scheme' method is based on the HTTP-Header `Authorization` in base 64 encoding, as described in RFC 2617 (Basic Authentication Scheme).

  ```
  If the user agent wishes to send the userid "Alibaba" and password "open sesame",
  it will use the following header field:
  > Authorization: Basic QWxpYmFiYTpvcGVuIHNlc2FtZQ==
  ```

  > **Note**
  >
  > The `WWW-Authenticate` `[p 686]` header can be valued with this method.

  Voir aussi*HTTP Authorization header policy* *[p 380]*

- 'Standard Authentication Scheme' is based on the HTTP `Request`. User and password are extracted from request parameters. For more information on request parameters, see Parameters [p 634] section.

  For more information on this authentication scheme, see `Directory.authenticateUserFromLoginPassword`[API].

- The 'REST Forward Authentication Scheme' is used only when calling a REST service from a user service [p 584], that reuses the current authenticated session.

  For more information, see Implementing a user service [p 595] making a call to REST data services [p 602].

- 'Specific authentication Scheme' is based on the HTTP `Request`. For example, an implementation can extract a password-digest or a ticket from the HTTP `Request`. See `Directory.authenticateUserFromHttpRequest`[API] for more information.

- 'Anonymous authentication Scheme' is used only to access the REST services handling the authentication operations. The credentials acquisition, password changes, etc. imply that the user cannot be known yet.

Global permissions

Global access permissions can be independently defined for the REST built-in services access. For more information see Global permissions [p 407].

Lookup mechanism

Because EBX offers several authentication methods, a lookup mechanism based on conditions was set to know which method should be applied for a given request. The method application conditions are evaluated according to the authentication scheme priority. If the conditions are not satisfied, the server evaluates the next method. The following table presents the available authentication methods for each supported protocol and their application conditions. They are ordered from the highest priority to the lowest.

Operation / Protocol	Authentication methods and application conditions
REST / HTTP	Token [p 676] • The HTTP request must hold an `Authorization` header. • `Authorization` header value must start with the word `EBX`. • No `login` is provided in the URL parameters. Basic [p 676] • The HTTP request must hold an `Authorization` header. • `Authorization` header value must start with the word `Basic`. • No `login` is provided in the URL parameters. Standard [p 676] • The HTTP request must not hold an `Authorization` header. • A `login` and a `password` are provided in the URL parameters. Rest forward [p 676] • The HTTP request must not contain an `Authorization` header. • No `login` is provided in the URL parameters. Specific [p 676] • The HTTP request must not satisfy the conditions of the previous authentication methods. Anonymous [p 677] • None of the previous authentication methods can be applied. • The requested REST service is handling an authentication operation.

In case of multiple authentication methods present in the same request, EBX will return an HTTP code `401 Unauthorized`.

98.5 **Monitoring**

Data service events can be monitored through the log category `ebx.dataServices`, as declared in the EBX main configuration file. For example, `ebx.log4j.category.log.dataServices= INFO, ebxFile:dataservices`.

Voir aussi

Configuring the EBX logs *[p 375]*

TIBCO EBX main configuration file *[p 369]*

98.6 **SOAP and REST comparative**

Operations	**SOAP**	**REST**
Data		
Select or count records (with filter and/or view publication)	X	X
Selector for possible enumeration values (with filter)		X
Insert, update or delete records	X	X
Select or count history records (with filter and/or view publication)		X
Select node values from dataset	X	X
Update node value from dataset		X
Get table or dataset changes between dataspaces or snapshots	X	
Refresh a replication unit	X	
Get credentials for records	X	
Dataspaces		
Create, close, merge a dataspace	X	
Create, close a snapshot	X	
Validate a dataspace or a snapshot	X	
Validate a dataset	X	
Locking a dataspace	X	
Workflow		
Start, resume or end a workflow	X	
Administration		
Manage the default directory content 'Users', 'Roles'... tables.	X	X
Open, close the user interface	X	X

Operations	SOAP	REST
Select, insert, update, delete operations for administration dataset		X
Select the system information	X	X
Other		
Develop web services from the Java API		X (*)

(*) See REST Toolkit [p 745] for more information.

98.7 Limitations

Date, time & dateTime format

Data services only support the following date and time formats:

Type	Format	Example
xs:date	yyyy-MM-dd	2007-12-31
xs:time	HH:mm:ss or HH:mm:ss.SSS	11:55:00
xs:dateTime	yyyy-MM-ddTHH:mm:ss or yyyy-MM-ddTHH:mm:ss.SSS	2007-12-31T11:55:00

JMS

- JMS protocol is not supported.

CHAPITRE **99**

Built-in RESTful services

Ce chapitre contient les sections suivantes :

1. Introduction
2. Request
3. Response
4. Administration operations
5. Token authentication operations
6. Data operations
7. Limitations

99.1 Introduction

The architecture used is called ROA (Resource-Oriented Architecture), it can be an alternative to SOA (Service-Oriented Architecture). The chosen resources are readable and/or writable by third-party systems, according to the request content.

The HATEOAS approach of the built-in RESTful services also allows to experience an intuitive and straightforward navigation, which implies that the data details could be obtained through a link.

> **Note**
>
> All operations are stateless.

99.2 Request

This chapter describes the elements to use in order to build a conform REST request, such as: the HTTP method, the URL format, the header fields and the message body.

Voir aussi

Interactions *[p 675]*

Security *[p 676]*

HTTP method

Considered HTTP methods for built-in RESTful services, are:

- `GET`: used to select master data defined in the URL (the URL size limit depends on the application server or on the browser, that must be lower than or equal to 2KB).
- `POST`: used to insert one or more records in a table or to select the master data defined in the URL (the size limit is 2MB or more depending on the application server. Each parameter is limited to a value containing 1024 characters).
- `PUT`: used to update the master data defined in the URL.
- `DELETE`: used to delete either the record defined in the URL or multiple records defined with the table URL and the record table in the message body.

URL

REST URL contains:

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/{categoryVersion}/
{specificPath}[:{extendedAction}]?{queryParameters}
```

Where:

- `<ebx-dataservices>` corresponds to the 'ebx-dataservices.war' web application's path. The path is composed by multiple, or none, URI segments followed by the web application's name.
- `{category}` corresponds to the operation category [p 683].
- `{categoryVersion}` corresponds to the category version: current value is `v1`.
- `{specificPath}` corresponds to a specific path inside the category.
- `{extendedAction}` corresponds to the extended action name (optional).
- `{queryParameters}` corresponds to common [p 685] or dedicated operation parameters passed by the URL.

Operation category

It specializes the operation, it is added in the path of the URL in `{category}` and it takes one of the following values:

`admin`	Administration operations reserved to administrators. For more information, see: Administration operations [p 688].
`auth`	Manage token authentication method. For more information, see: Token authentication operations [p 690] and Token Authentication Scheme [p 676].
`data`	Lists dataset content, requests a table, a record or a field record content, including modified operations on dataset node, table, record and record field. For more information, see: Data operations [p 693].
`history`	Lists history dataset content, requests a history table, a history of a record or a history record. For more information, see: Data operations [p 693]. **Voir aussi***Historique [p 273]*

Header fields

These header field definitions are used by TIBCO EBX.

`Accept`	Used to specify content types by order of preference to be used in the response, the first supported one will be chosen and specified in the response header `Content-Type`. Currently, the only supported one is `application/json`. If none is supported, the result depends on the `ebx.dataservices.rest.request.checkAccept` property: • If `true`, an HTTP error response is returned with code `406`. • If `false`, the response is returned with the default content type, that is `application/json`. **Voir aussi***Configuring data services* *[p 378]*
`Accept-Language`	Used for specifying the preferred locale for the response. The supported locales are defined in the schema model. If none of the preferred locale are supported, the default locale for the current model is used.
`Authorization`	Used for 'Basic Authentication Scheme' and 'Token Authentication Scheme' methods, otherwise the request is rejected. **Voir aussi***Authentication* *[p 676]*
`Content-Type`	Used to specify the request body media type. The supported types are `application/json` and `application/x-www-form-urlencoded`. The request value is checked and if it is not supported, then an HTTP error message is returned with the code `415` *(Unsupported media type)*. **Voir aussi***Configuring data services* *[p 378]*
`X-Requested-With`	If present and in case of authentication failure, prevents the addition of the `WWW-Authenticate` header in the response. **Voir aussi***Response header `WWW-Authenticate`* *[p 686]*

See RFC2616 for more information about HTTP Header Field Definitions.

Common parameters

These optional parameters are available for all data service operations.

Parameter	Description
`disableRedirectionToLastBroadcast`	This parameter only has impact on a D3 architecture. If `true`, access to a delivery dataspace on a D3 primary node is not redirected to the last broadcast snapshot. Otherwise, access to such a dataspace is always redirected to the last broadcast snapshot. If the specified dataspace is not a delivery dataspace on a D3 primary node, this parameter is ignored. `Boolean` type value. If this parameter is not present, the default is `false` (redirection to a D3 master enabled), unless the configuration property ebx.dataservices.disableRedirectionToLastBroadcast.default [p 378] has been set. **Voir aussi***Primary node* *[p 451]*
`indent`	Specifies if the response should be indented, to be easier to read for a human. `Boolean` type, default value is `false`.

Message body

It contains the request data using the JSON format, see JSON Request body [p 720].

> **Note**
>
> Requests may define a message body only when using `POST` or `PUT` HTTP methods.

99.3 **Response**

This chapter describes the responses returned by built-in RESTful services.

- See Exception handling [p 675] for details on standard error handling (where the HTTP code is greater than or equal to `300`).

Header fields

These header field definitions are used by EBX.

`Content-Language`	Indicates the locale used in the response for labels and descriptions.
`Content-Type`	Indicates the response body content type.
`Location`	If a new record has been successfully inserted, the query URL for this record is returned by this field.
`WWW-Authenticate`	This header field is added to the HTTP response when authentication fails with the 401 *(Unauthorized)* HTTP code. Its value consists of a list with at least one authentication method applicable to the request URI. It is present if and only if the following conditions are verified: • the 'Basic Authentication Scheme' method is enabled and • the `X-Requested-With` HTTP header is not present. If the client is able to interpret the authentication method, it is possible to resubmit the request providing the appropriate credentials. The administration property ebx.dataservices.rest.auth.tryBasicAuthentication [p 378] must be set to `true`. **Voir aussi** *Request header `X-Requested-With`* *[p 684]* *Authentication* *[p 676]*

HTTP codes

HTTP code	Description
`200` *(OK)*	The request has been successfully handled.
`201` *(Created)*	A new record has been created, in this case, the header field `Location` is returned with its resource URL.
`204` *(No content)*	The request has been successfully handled but no response body is returned.
`400` *(Bad request)*	the request URL or body is not well-formed or contains invalid content.
`401` *(Unauthorized)*	Authentication has failed.
`403` *(Forbidden)*	Permission was denied to read or modify the specified resource for the authenticated user. This error is also returned when the user: • is not allowed to modify a field mentioned in the request message body. • is not allowed to access the REST connector. For more details, see Global permissions [p 677].
`404` *(Not found)*	The resource specified in the URL cannot be found.
`406` *(Not acceptable)*	Content type defined in the request's `Accept` parameter is not supported. This error can be returned only if the EBX property `ebx.rest.request.checkAccept` is set to `true`.
`409` *(Conflict)*	A concurrent modification has occurred. **Voir aussi***Optimistic locking [p 713]*
`415` *(Unsupported media type)*	The request content is not supported, the request header value `Content-Type` is not supported by the operation.
`422` *(Unprocessable entity)*	The new resource's content cannot be accepted for semantic reasons.
`500` *(Internal error)*	Unexpected error thrown by the application. Error details can usually be found in EBX logs.

Message body

The response body content's format depends on the HTTP code value:

- HTTP codes from `200` included to `300` excluded: the content format depends on the associated request (JSON [p 723] samples).

  With the exception of code `204` *(No content)*.
- HTTP codes greater than or equal to `300`: the content describes the error. See JSON [p 675] for details on the format.

99.4 Administration operations

Administration operations are related to:

- the administration category.
- the administration dataspaces accessible through the `data` category.

> **Note**
>
> administration category and administration dataspaces can only be used by administrators.

Directory operations

The EBX default directory configuration is manageable with built-in RESTful services. The users and roles tables, the mailing lists and other objects are concerned. For more information, see Users and roles directory [p 423].

> **Note**
>
> Triggers are present on the directory's tables, ensuring the data consistency.

The URL format is:

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/Bebx-directory/ebx-directory
```

Directory configuration operations

The EBX default directory configuration is manageable like dataset nodes. It can be accessed and modified through the `data` category operations. Each field is self-described when metadata is requested.

See select [p 693] and update [p 706] operations for more information.

Mailing lists operations

There are two default mailing lists that can be configured in the EBX directory: one for everybody and one for administrators. These lists can be handled like dataset nodes through the `data` category operations.

See select [p 693] and update [p 706] operations for more information.

Directory users operations

Users can be manipulated like records of the `data` category using the operations of the latter. For security purposes, an administrator cannot delete himself. The user's salutation must be chosen among the ones available in the 'salutations' table.

See select [p 693], update [p 706], insert [p 701] and delete [p 708] operations for more information.

Directory roles operations

Roles are records of the `data` category and can be managed with its operations. EBX roles are assigned to users through the 'usersRoles' association table. 'usersRoles' is automatically fed when the directory is administered through the user interface. However, it is not the case through data services and role assignments require manual operations. Roles inclusions are specified in the 'rolesInclusions'

association table. As for the 'usersRoles' table, the management of roles inclusions requires manual operations. Each table is self-descriptive when metadata is requested.

See select [p 693], update [p 706], insert [p 701] and delete [p 708] operations for more information.

User interface operations

The EBX user interface can be opened or closed to users for maintenance needs. Handled information is similar to what is contained in the UI tab 'Administration' > 'User interface configuration' > 'Advanced perspective' > 'Graphical interface configuration' > 'Application locking'.

URL format is:

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/Bebx-manager/ebx-manager/
domain/toolStatus
```

Voir aussi*Application locking [p 412]*

Retrieve user interface state

User interface status and the unavailability message are accessible like dataset nodes.

See Select operation [p 693] and the JSON [p 730] example, for more information.

Open or close user interface

User interface status and the unavailability message can be modified like dataset nodes using the update operation. To open the user interface set the content of `toolStatus` to `true`, or to `false` to close it.

See Update operation [p 706] and the JSON [p 723] examples, for more information.

System information operation

This operation returns system information on the EBX server. This is accepted for `GET` and `POST` HTTP methods. Warning: no update will be possible in the `POST` HTTP method because the request body is ignored. The information returned is the same as the information contained in the log header `kernel.log` or in the UI tab 'Administration' > 'System Information'. The response contains several keys, labels, and values representing the configuration and status of EBX. The mode of representation of the response may be flat or hierarchical.

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/admin/v1/systemInformation
```

Voir aussi

TIBCO EBX main configuration file [p 369]

Repository administration [p 396]

Parameters

The following parameter is applicable.

Parameter	Description
`systemInformationMode`	Specifies the returned mode: • `flat`: A flat representation under the following information groups: `bootInfoEBX`, `repositoryInfo` and `bootInfoVM`. • `hierarchical`: A hierarchical representation. `String` type, default value is `flat`.

HTTP codes

HTTP code	Description
`200` *(OK)*	The system information was successfully returned.
`400` *(Bad request)*	The request is not correct, it contains one of the following errors: • the HTTP method is not `GET` nor `POST`, • the HTTP parameter `systemInformationMode` is not correct, • the operation is not supported. • the request path is invalid.
`403` *(Forbidden)*	The user is not an administrator.

Response body

It is returned, if and only if, the HTTP code is `200 (OK)`. The content structure depends on the provided parameter `systemInformationMode` or its default value.

See the JSON [p 724] example of the flat representation.

See the JSON [p 724] example of the hierarchical representation.

99.5 Token authentication operations

These operations allow to create or revoke an authentication token. Authentication tokens have a timeout period. If a token is not used to access the EBX server within this period, it will automatically be revoked. This timeout period is refreshed on each access to EBX server.

> **Note**
>
> The token timeout is modifiable through the administration property ebx.dataservices.rest.auth.token.timeout [p 378] (the default value is 30 minutes).

Create token operation

This operation requires using the `POST` HTTP method with a request containing the user's credentials and, optionally, session parameters [p 675].

URL format is:

`http[s]://<host>[:<port>]/<ebx-dataservices>/rest/auth/v1/token:create`

Message body

A message body must be defined in the HTTP request. It necessarily contains one of the following set of data:

- A `login` and a `password` value. Both JSON attributes are mandatory and of `String` types.

  See `Directory.authenticateUserFromLoginPassword`[API] for more information.

- The `specific` JSON attribute set to `true`. When activated, this flag allows to performed a user authentication against the whole HTTP request. Warning, even if `login` and `password` attributes are defined in the JSON request's body, setting `specific` to `true` lead to a specific user authentication.

  See `Directory.authenticateUserFromHttpRequest`[API] for more information.

See the JSON [p 720] examples of a token creation request.

HTTP codes

HTTP code	Description
`200` *(OK)*	The token was successfully created.
`400` *(Bad request)*	For one of the following reasons: • the syntax is not correct, • the HTTP method is not `POST`, • the operation is not supported.
`401` *(Unauthorized)*	For one of the following reasons: • The `login` and/or `password` is/are incorrect. • Authentication data for other methods is defined.
`422` *(Unprocessable entity)*	For one of the following reasons: • `PasswordMustChange`: The password must be changed (only available with the default directory). See Change password operation [p 692]. • `RestrictedAccess`: User access is closed for maintenance or other actions (reserved to administrators).

Response body

If the HTTP code is `200 (OK)`, the body holds the token value and its type.

See the JSON [p 725] example of a token creation response.

The token can later be used to authenticate a user by setting the HTTP-Header `Authorization` accordingly.

Voir aussi *'Token authentication Scheme' method [p 676]*

Change password operation

This operation modifies the password of an existing user account. It can be used in an authenticated context: `login` parameter, if present, is checked against the current session or taken from it, if absent. It could also be used in an unauthenticated context, for example when the Create token operation [p 690] aborts with the HTTP code `422` *(Unprocessable entity)* with reason: `PasswordMustChange`.

It requires the use of:

- the EBX default directory
- the `POST` HTTP method
- the message body containing the structure specified below

URL format is:

`http[s]://<host>[:<port>]/<ebx-dataservices>/rest/auth/v1/user:changePassword`

Message body

The message body must be defined in the request. It necessarily contains a `password` and a `passwordNew`, the `login` is optional (all are `String`).

See the JSON [p 720] example of a password change and token creation request.

HTTP codes

HTTP code	Description
`204` *(No content)*	The password has been changed.
`400` *(Bad request)*	For one of the following reasons: • the EBX default directory is required, • the syntax is not correct, • the HTTP method is not `POST`, • the provided `login` and the user's session one mismatch • the operation is not supported.
`401` *(Unauthorized)*	For the following reason: • The `login` and/or `password` is/are incorrect.
`422` *(Unprocessable entity)*	For one of the following reasons: • `PasswordChangeAbort`: `passwordNew` is empty. • `PasswordChangeAbort`: `passwordNew` is equal to `password`. • `PasswordChangeAbort`: `password` is incorrect. • `RestrictedAccess`: User access is closed for maintenance or other actions (reserved to administrators).

Response body

If HTTP code `204 (No content)` is returned, then the password has been modified.

Revoke token operation

This operation requires using the `POST` HTTP method. No message body is needed.

URL format is:

`http[s]://<host>[:<port>]/<ebx-dataservices>/rest/auth/v1/token:revoke`

Header fields

`Authorization`	This field is required, `tokenType` and `accessToken` fields must have the values returned from the "token create" operation. `> Authorization: <tokenType> <accessToken>`

HTTP codes

HTTP code	Description
`204` *(No content)*	The token has been revoked successfully.
`400` *(Bad request)*	For one of the following reasons: • the configuration is not activated, • the syntax is incorrect, • the HTTP method is not `POST`, • the operation is not supported.
`401` *(Unauthorized)*	Authentication has failed.

99.6 Data operations

Operations from the `data` operation category concern the datasets, the dataset fields, tables, records or record fields.

Operations from the `history` category and concern historized content from datasets, tables, records or the record fields.

Select operation

Select operation may use one of the following methods:

- `GET` HTTP method,
- `POST` HTTP method without message body or
- `POST` HTTP method with message body and optionally session parameters [p 675].

URL formats are:

- **Dataset tree**, depending on operation category [p 683]:

  The `data` category returns the hierarchy of the selected dataset, this includes group and table nodes.

The `history` category returns the hierarchy of the selected history dataset, this includes the pruned groups for history table nodes only.

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
{dataset}
```

> **Note**
>
> Terminal nodes and sub-nodes are not included.

- **Dataset node**: the `data` category returns the terminal nodes contained in the selected node.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}
  ```

  > **Note**
  >
  > Not applicable with the `history` category.

- **Table**, depending on operation category [p 683]:

  the `data` category returns the table content and/or metadata, current page records and URLs for pagination.

  The `history` category returns the history table content and/or metadata, current page records and URLs for pagination.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
  {dataset}/{pathInDataset}
  ```

  Voir aussi*Count operation [p 710]*

- **Record**, depending on operation category [p 683]:

  the `data` category returns the record content and/or metadata.

  The `history` category returns history record content and/or metadata.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
  {dataset}/{pathInDataset}/{encodedPrimaryKey}
  ```

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
  {dataset}/{pathInDataset}?primaryKey={xpathExpression}
  ```

  > **Note**
  >
  > The record access by the primary key (`primaryKey` parameter) is limited to its root node. It is recommended to use the encoded primary key, available in the `details` field in order to override this limitation. Similarly, for a history record, use the encoded primary key, available in the `historyDetails` field.

- **Field**, depending on operation category [p 683]:

  the `data` category returns the field record content where structure depends on its type.

The `history` category returns the field history record content where structure depends on its type.

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
{dataset}/{pathInDataset}/{encodedPrimaryKey}/{pathInRecord}
```

> **Note**
>
> The field must be either an association node, a selection node, a terminal node or above.

Where:

- `{category}` corresponds to the operation category [p 683].
- `{dataspace}` corresponds to `B` followed by the dataspace identifier or to `V` followed by the snapshot identifier.
- `{dataset}` corresponds to the dataset identifier.
- `{pathInDataset}` corresponds to the path of the dataset node, that can be a group node or a table node.
- `{encodedPrimaryKey}` corresponds to the encoded representation of the primary key.

  Voir aussi *`RESTEncodingHelper`*[API]

- `{xpathExpression}` corresponds to the record primary key, using the XPath expression.
- `{pathInRecord}` corresponds to the path starting from the table node.

Parameters

The following parameters are applicable to the select operation.

Parameter	Description
`includeContent`	Includes the `content` field with the content corresponding to the selection. `Boolean` type, default value is `true`.
`includeDetails`	Includes the `details` field in the metadata and the response, for each indirect reachable resource. The returned value corresponds to its URL resource. Type `Boolean`, default value is `true`. **Voir aussi** *includeMeta* [p 696]
`includeHistory`	Includes those fields for a historized content: • The `history` property in the metadata. The returned value corresponds to a boolean value. • The `historyDetails` property in the content and for each indirectly reachable resource. This point is coupled to the includeDetails [p 696] parameter. The returned value corresponds to its URL resource. `Boolean` type, default value is `false`. **Note** The `includeHistory` parameter is ignored in the `history` category, the default value is `true`. **Voir aussi** *includeMeta* [p 696] *includeDetails* [p 696] *`AdaptationTable.getHistory`* API
`includeLabel`	Includes the `label` field associated with each simple type `content`. Possible values are: • `yes`: the `label` is included for the foreign key, enumeration, record and selector values. • `all`: the `label` field is included, as for the `yes` value and also for the Content of simple type [p 737]. **Voir aussi** *`SchemaNode.displayOccurrence`* API • `no`: the `label` field is not included (integration use case). `String` type, default value is `yes`. **Note** The `label` field is not included if it is equal to the `content` field.
`includeMeta`	Includes the `meta` field corresponding to the description of the structure returned in the `content` field. `Boolean` type, default value is `false`.

Parameter	Description
	Voir aussi *includeHistory* [p 696] *includeDetails* [p 696]
`includeMergeInfo`	Includes the `merge_info` corresponding to a field of the technical data of an history transaction and has a potentially high access cost. `Boolean` type, default value is `true`. **Note** This parameter is ignored with the `data` category. **Voir aussi***Accs REST aux tables historises* [p 327]
`includeSelector`	Includes the `selector` field in the response, for each indirect reachable resource. The returned value corresponds to its URL resource. Type `Boolean`, default value is `true`. **Voir aussi***selector* [p 700]
`includeSortCriteria`	Includes the `sortCriteria` field corresponding to the list of sort criteria applied. The sort criteria parameters are added by using: • sort [p 699] • sortOnLabel [p 699] • viewPublication [p 700] `Boolean` type, default value is `false`. Example JSON [p 733]
`includeTechnicals`	Includes the internal technical data. `Boolean` type, default value is `false`. **Voir aussi***Technical data* [p 742] **Note** This parameter is ignored with the `history` category.
`includeValidation`	Includes the validation report corresponding to the selection. `Boolean` type, default value is `false`. **Note** This parameter is ignored with the `history` category. **Voir aussi***includeDetails* [p 696]

Table parameters

The following parameters are applicable to tables, associations and selection nodes.

Parameter	Description
`filter`	XPath predicate [p 249] expression defines the field values to which the request is applied. If empty, all records will be retrieved. `String` type value. **Note** The history code operation value is usable with `ebx-operationCode` path field from the `meta` section associated with this field.
`historyMode`	Specifies the filter context applied on table. `String` type, possible values are: • `CurrentDataSpaceOnly`: history in current dataspace • `CurrentDataSpaceAndAncestors`: history in current dataspace and ancestors • `CurrentDataSpaceAndMergedChildren`: history in current dataspace and merged children • `AllDataSpaces`: history in all dataspaces The default value is `CurrentDataSpaceOnly`. **Voir aussi** *`history` [p 683]* **Note** This parameter is ignored with the `data` category.
`includeOcculting`	Includes the records in occulting mode. `Boolean` type, default value is `false`.
`primaryKey`	Search a record by a primary key, using the XPath expression. The XPath predicate [p 249] expression should only contain field(s) of the primary key and all of them. Fields are separated by the operator `and`. A field is represented by one of the following possibilities according to its simple type: • For the `date`, `time` or `dateTime` types: use the `date-equal(path, value)` • For other types: indicate the `path`, the `=` operator and the `value`. Example with a composed primary key: `./pk1i=1 and date-equal(./pk2d,'2015-11-13')` The response will only contain the corresponding record, otherwise an error is returned. Consequently, the other table parameters are ignored (as filter [p 698], viewPublication [p 700], sort [p 699], etc.) `String` type value.
`pageFirstRecordFilter`	*Deprecated since version 5.9.0*, replaced by `pageRecordFilter`
`pageRecordFilter`	Specifies the record XPath predicate [p 249] expression filter of the page. `String` type value.

Parameter	Description
`pageAction`	Specifies the pagination action to perform from page defined by `pageRecordFilter`. `String` type, possible values are: • `first` • `previous` • `next` • `last` **Voir aussi** *`RequestPagination`*[API]
`pageSize`	Specifies the number of records per page. `Integer` type, default value is based on the user preferences. `String` type, the `unbounded` value can be defined to return all records. Only for this case, no pagination context is returned.

`sort`	Specifies that the operation result will be sorted according to the specified criteria. The criteria are composed of one or more criteria, the result will be sorted by priority from the left. A criterion is composed of the field path and, optionally, the sorting order (ascending or descending, on value or on label). This parameter can be combined with:
	1. the sortOnLabel [p 699] parameter as a new criteria added after the `sort`.
	2. the viewPublication [p 700] parameter as a new criteria added after the `sort`.
	The value structure is as follows:
	`<path1>:<order>;...;<pathN>:<order>`
	Where:
	• `<path1>` corresponds to the field path in priority `1`.
	• `<order>` corresponds to the sorting order, with one of the following values:
	  • `asc`: ascending order on value (default),
	  • `desc`: descending order on value,
	  • `lasc`: ascending order on label,
	  • `ldesc`: descending order on label.
	`String` type, the default value orders according to the primary key fields (ascending order on value).
	Note
	The history code operation value is usable with the `ebx-operationCode` path field from the `meta` section associated with this field.
	Voir aussi *`Request.setSortCriteria`*[API]

`sortOnLabel`	Specifies that the operation result will be sorted according to the record label. This parameter can be combined with: 1. the sort [p 699] parameter as a new criteria added before the `sortOnLabel`. 2. the viewPublication [p 700] parameter as a new criteria added after the `sortOnLabel`. The value structure is as follows: `<order>`

Parameter	Description

	Where:
	• `<order>` corresponds to the sorting order, with one of the following values:
	  • `lasc`: ascending order on label,
	  • `ldesc`: descending order on label.
	The behavior of this parameter is described in the section defaultLabel [p 519].
	`String` type value.
	Voir aussi*Limitation [p 716]*

`viewPublication`	Specifies the name of the published view. This parameter can be combined with: 1. the filter [p 698] parameter as the logical `and` operation. 2. the sort [p 699] parameter as a new criteria added before the `viewPublication`. 3. the sortOnLabel [p 699] parameter as new criteria added before the `viewPublication`. The behavior of this parameter is described in the section EBX as a Web Component [p 215]. `String` type value.

Selector parameters

The following parameters are only applicable to fields that return an enumeration, foreign key or `osd:resource` (Example JSON [p 740]). By default, a pagination mechanism is enabled.

Parameter	Description
`selector`	Specifies whether: • `true`: returns all possible values (includes their labels) • `false`: returns the current values for the current field. `Boolean` type, default value is `false`. **Note** This parameter is ignored with the `history` category.
`firstElementIndex`	Specifies the index of the first element returned by the selector. Must be an integer higher than or equal to `0`. `Integer` type, default value is `0`.
`pageSize`	Specifies the number of elements per page. `Integer` type, default value is based on the user preferences. `String` type, the `unbounded` value can be defined to return all values. Only for this case, no pagination context is returned.
`selectorFilter`	Specifies the filter of the selector. `String` type value, the syntax complies with the Recherche textuelle [p 126].

HTTP codes

HTTP code	Description
`200` *(OK)*	The selected resource is successfully retrieved.
`400` *(Bad request)*	The request is incorrect. This occurs when: • The selected field in a record or a dataset is sub-terminal, • The XPath predicate of the `filter` parameter is malformed or contains unfilterable nodes. • The XPath predicate of the `primaryKey` parameter is malformed or is not a record primary key. • The sort criteria of the `sort` parameter have an invalid syntax or contain unsortable nodes. • `pageAction` parameter value is not included in allowed values, or `pageRecordFilter` is malformed or non-existent when selecting next or previous page. • `pageSize` parameter value is below 2, or is a string different from `unbounded`. • The table view for the `viewPublication` parameter is either hierarchical, non-existent or non-published. • The `selector` parameter is used for a non-enumerated node, or the `firstElementIndex` is negative, higher than or equal to the number of values.
`403` *(Forbidden)*	The selected resource is hidden for the authenticated user.
`404` *(Not found)*	The selected resource is not found.

Response body

After a successful dataset, table, record or field selection, the result is returned in the response body. The content depends on the provided parameters and selected data.

Example: JSON [p 725].

Insert operation

Insert operation uses the `POST` HTTP method. A body message is required to specify data. This operation supports the insertion of one or more records in a single transaction. Moreover, it is also possible to update record(s) through parameterization.

- **Record**: insert a new record or modify an existing one in the selected table.
- **Record table**: insert or modify one or more records in the selected table, while securing a consistent answer. Operations are executed sequentially, in the order defined on the client side. When an error occurs during a table operation, all updates are cancelled and the client receives an error message with detailed information.

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
{pathInDataset}
```

Where:

- `{dataspace}` corresponds to `B` followed by the dataspace identifier or to `V` followed by the snapshot identifier.

- `{dataset}` corresponds to the dataset identifier.
- `{pathInDataset}` corresponds to the path of the table node.

Parameters

The following parameters are applicable with the insert operation.

Parameter	Description
`includeDetails`	Includes the `details` field in the answer for access to the details of the data. The returned value corresponds to its URL resources. Type `Boolean`, the default value is `false`. **Note** Only applicable on the **record table**.
`includeForeignKey`	Includes the `foreignKey` field in the answer for each record. The returned value corresponds to the value of a foreign key field that was referencing this record. `Boolean` type, the default value is `false`. **Note** Only applicable on the **record table**.
`includeLabel`	Includes the `label` field in the answer for each record. Possible values are: • `yes`: the `label` field is included. • `no`: the `label` field is not included (use case: integration). `String` type, the default value is `no`. **Note** Only applicable on the **record table**.
`updateOrInsert`	Specifies the behavior when the record to insert already exists: • If `true`: the existing record is updated with new data. For a request on a **record table**, the `code` field is added to the report in order to specify if this is an insert `201` or an update `204`. • If `false` (default value): a client error is returned and the operation is aborted. `Boolean` type value.
`byDelta`	Specifies the behavior for setting value of nodes that are not defined in the request body. This is described in the Update modes [p 742] section. `Boolean` type, the default value is `true`. **Note** Applicable on record in update mode and if the updateOrInsert [p 703] parameter is `true`.
`blockingConstraintsDisabled`	Specifies whether blocking constraints are ignored, if so, the operation is committed regardless of the validation error created, otherwise, the operation would be aborted. `Boolean` type, default value is `false`.

Parameter	Description
	See Blocking and non-blocking constraints [p 546] for more information.

Message body

The request must define a message body. The format depends on the inserted object type:

- **Record**: similar to the select operation of a record but without the record's header (example JSON [p 721]).
- **Record table**: Similar to the select operation on a table but without the pagination information (example JSON [p 722]).

Voir aussi*Inheritance [p 714]*

HTTP codes

HTTP code	Description
`200` *(OK)*	If the request relates to a **record table**. The insert request was applied successfully, an optional report is returned in the response body.
`201` *(Created)*	If the request relates to a **record**. A new record has been created, in this case, the header field `Location` is returned with its resource URL.
`204` *(No content)*	If the request relates to a **record**. Only available if `updateOrInsert` is `true`, an existing record has been successfully updated, in this case, the header field `Location` is returned with its resource URL.
`400` *(Bad request)*	The request is incorrect. This occurs when the body message structure does not comply with what was mentioned in Message body [p 704].
`403` *(Forbidden)*	Authenticated user is not allowed to create a record or the request body contains a read-only field.
`404` *(Not found)*	The selected resource is not found.
`409` *(Conflict)*	Concurrent modification, only available if `updateOrInsert` is `true`, the Optimistic locking [p 713] is activated and the content has changed in the meantime, it must be reloaded before update.
`422` *(Unprocessable entity)*	The request cannot be processed. This occurs when: • A blocking validation error occurs (only available if `blockingConstraintsDisabled` is `false`). • The record cannot be inserted because a record with the same primary key already exists (only available if `updateOrInsert` is `false`). • The record cannot be inserted because the definition of the primary key is either non-existent or incomplete. • The record cannot be updated because the value of the primary key cannot be modified.

Response body

The response body format depends on the inserted object type:

- **Record**: is empty if the operation was executed successfully. The header field `Location` is returned with its URL resource.
- **Record table**: (optional) contains a table of element(s), corresponding to the insert operation report (example JSON [p 740]). This report is automatically included in the response body, if at least one of the following options is set:
  - `includeForeignKey`
  - `includeLabel`
  - `includeDetails`

Voir aussi*Inheritance [p 714]*

Update operation

This operation allows the modification of a single dataset or record. The `PUT` HTTP method must be used. Available URL formats are:

- **Dataset node**: modifies the values of terminal nodes contained in the selected node.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}
  ```

- **Record**: modifies the content of selected record.

  > **Note**
  >
  > Also available for `POST` HTTP methods. In this case, the URL must correspond to the table by setting the parameter `updateOrInsert` to `true`.

- **Field**: modifies the field content.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}/{encodedPrimaryKey}/{pathInRecord}
  ```

  > **Note**
  >
  > The field must be either a terminal node or above.

Where:

- `{dataspace}` corresponds to `B` followed by the dataspace identifier or to `V` followed by the snapshot identifier.
- `{dataset}` corresponds to the dataset identifier.
- `{pathInDataset}` corresponds to the path of the dataset node:
  - For dataset node operations, this must be any terminal node or above except table node,
  - For record and field operations, this corresponds to the table node.
- `{encodedPrimaryKey}` corresponds to the encoded representation of the primary key.

  Voir aussi*`RESTEncodingHelper`*[API]

- `{pathInRecord}` corresponds to the path starting from the table node.

Parameters

Here are the parameters applicable with the update operation.

Parameter	Description
`blockingConstraintsDisabled`	Specifies whether blocking constraints are ignored, if so, the operation is committed regardless of the validation error created, otherwise, the operation would be aborted. `Boolean` type, default value is `false`. See Blocking and non-blocking constraints [p 546] for more information.
`byDelta`	Specifies the behavior for setting value of nodes that are not defined in the request body. This is described in the Update modes [p 742] section. `Boolean` type, the default value is `true`.
`checkNotChangedSinceLastUpdateDate`	Timestamp in datetime format used to ensure that the record has not been modified since the last read. Also see the Optimistic locking [p 713] section. `DateTime` type value.

Message body

The request must define a message body.

The structure is the same as for:

- the dataset node (sample JSON [p 720]),
- the record (sample JSON [p 721]),
- the record fields (sample JSON [p 721]),

depending on the updated scope, by only keeping the `content` entry.

Voir aussi*Inheritance [p 714]*

HTTP codes

HTTP code	Description
`204` *(No content)*	The record, field or dataset node has been successfully updated.
`400` *(Bad request)*	The request is incorrect. This occurs when the body request structure does not comply.
`403` *(Forbidden)*	Authenticated user is not allowed to update the specified resource or the request body contains a read-only field.
`404` *(Not found)*	The selected resource is not found.
`409` *(Conflict)*	Concurrent modification, the Optimistic locking [p 713] is activated and the content has changed in the meantime, it must be reloaded before the update.
`422` *(Unprocessable entity)*	The request cannot be processed. This occurs when: • A blocking validation error occurs (only available if `blockingConstraintsDisabled` is `false`). • The record cannot be updated because the value of the primary key cannot be modified.

Delete operation

The operation uses the `DELETE` HTTP method.

Two URL formats are available:

- **Record**: delete a record specified in the URL.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}/{encodedPrimaryKey}
  ```

- **Record table**: deletes several records in the specified table, while providing a consistent answer. This mode requires a body message containing a record table. The deletions are executed sequentially, according to the order defined in the table. When an error occurs during a table operation, all deletions are cancelled and an error message is displayed with detailed information.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}
  ```

Where:

- `{dataspace}` corresponds to `B` followed by the dataspace identifier or to `V` followed by the snapshot identifier.
- `{dataset}` corresponds to the dataset identifier.
- `{pathInDataset}` corresponds to the path of the table node.
- `{encodedPrimaryKey}` corresponds to the encoded representation of the primary key.

Voir aussi *`RESTEncodingHelper`*[API]

In a child dataset context, this operation modifies the `inheritanceMode` property value of the record as follows:

- A record with inheritance mode set to `inherit` or `overwrite` becomes `occult`.
- A record with inheritance mode set to `occult` becomes `inherit` if the `inheritIfInOccultingMode` operation parameter is set to `true` or is undefined, otherwise there is no change.
- A record with inheritance mode set to `root` is simply deleted.

Voir aussi*Inheritance* *[p 714]*

Parameters

Here are the following parameters applicable with delete operation.

Parameter	Description
`includeOcculting`	Includes occulted records. `Boolean` type, the default value is `false`.
`inheritIfInOccultingMode`	*Deprecated since version 5.8.1* While it remains available for backward compatibility reasons, it will eventually be removed in a future version. Inherits the record if it is in occulting mode. `Boolean` type, the default value is `true`.
`checkNotChangedSinceLastUpdateDate`	Timestamp in datetime format used to ensure that the record has not been modified since the last read. Also see the Optimistic locking [p 713] section. `DateTime` type value.
`blockingConstraintsDisabled`	Specifies whether blocking constraints are ignored, if so, the operation is committed regardless of the validation error created, otherwise, the operation would be aborted. `Boolean` type, default value is `false`. See Blocking and non-blocking constraints [p 546] for more information.

Message body

The request must define a message body only when deleting several records:

- **Record table**: The message contains a table of elements related to a record, with for each element one of the following properties:
  - `details`: corresponds to the record URL, it is returned by the select operation.
  - `primaryKey`: corresponds to the primary key of the record, using the XPath expression.
  - `foreignKey`: corresponds to the value that a foreign key would have if it referred to a record.

  Voir aussi*`PrimaryKey`*[API]

  Example JSON [p 722].

HTTP codes

HTTP code	Description
`200` *(OK)*	The operation has been executed successfully. A report is returned in the response body.
`400`*(Bad request)*	The request is incorrect. This occurs when: • the structure of the message body does not comply with Message body [p 709]. • the message body contains a record table while the URL specifies a record.
`403` *(Forbidden)*	Authenticated user is not allowed to delete or occult the specified record.
`404` *(Not found)*	The selected record is not found. In the child dataset context, it should be necessary to use the `includeOcculting` parameter.
`409` *(Conflict)*	Concurrent modification, The Optimistic locking [p 713] is activated and the content has changed in the meantime, it must be reloaded before deleting the record. The parameter value `checkNotChangedSinceLastUpdateDate` exists but does not correspond to the actual last update date of the record.
`422` *(Unprocessable entity)*	Only available if `blockingConstraintsDisabled` is `false`, the operation fails because of a blocking validation error.

Response body

After a successful record deletion or occulting, a report is returned in the response body. It contains the number of deleted, occulted and inherited record(s).

Example JSON [p 741].

Count operation

Count operation may use one of the following methods:

- `GET` HTTP method,
- `POST` HTTP method without message body or
- `POST` HTTP method with message body but without `content` field on root.

The URL formats are:

- **Dataset node**: the `data` category returns the number of terminal nodes contained in the selected node.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/data/v1/{dataspace}/{dataset}/
  {pathInDataset}?count=true
  ```

  > **Note**
  >
  > Not applicable with the `history` category.

- **Table** depending on the operation category [p 683]:

  the `data` category returns the number of table records.

The `history` category returns the number of table history records.

```
http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
{dataset}/{pathInDataset}?count=true
```

- **Field** depending on the operation category [p 683]:

  the `data` category counts the record fields.

  The `history` category counts the history record field.

  ```
  http[s]://<host>[:<port>]/<ebx-dataservices>/rest/{category}/v1/{dataspace}/
  {dataset}/{pathInDataset}/{encodedPrimaryKey}/{pathInRecord}?count=true
  ```

  > **Note**
  >
  > The field must be either an association node, a selection node, a terminal node or above.

Where:

- `{category}` corresponds to the operation category [p 683].
- `{dataspace}` corresponds to `B` followed by the dataspace identifier or to `V` followed by the snapshot identifier.
- `{dataset}` corresponds to the dataset identifier.
- `{pathInDataset}` corresponds to the path of the dataset node, that can be a group node or a table node.
- `{encodedPrimaryKey}` corresponds to the encoded representation of the primary key.

  Voir aussi *`RESTEncodingHelper`*[API]

- `{pathInRecord}` corresponds to the path starting from the table node.

Parameters

The following parameters are applicable to the count operation.

Parameter	Description
`count`	This parameter is used to specify whether this is a count operation or a selection operation. `Boolean` type, default value is `false`.

Table parameters

The following parameters are applicable to tables, associations and selection nodes.

Parameter	Description
`filter`	XPath predicate [p 249] expression defines the field values to which the request is applied. If empty, all records will be considered. `String` type value. **Note** The history code operation value is usable with the `ebx-operationCode` path field from the `meta` section associated with this field.
`historyMode`	Specifies the filter context applied on table. `String` type, possible values are: • `CurrentDataSpaceOnly`: history in current dataspace • `CurrentDataSpaceAndAncestors`: history in current dataspace and ancestors • `CurrentDataSpaceAndMergedChildren`: history in current dataspace and merged children • `AllDataSpaces`: history in all dataspaces The default value is `CurrentDataSpaceOnly`. **Voir aussi** *`history`* *[p 683]* **Note** This parameter is ignored with the `data` category.
`includeOcculting`	Includes the records in occulting mode. `Boolean` type, default value is `false`.
`viewPublication`	Specifies the name of the published view to be considered during the count execution. This parameter can be combined with: • the filter [p 712] parameter as the logical `and` operation. The behavior of this parameter is described in the section EBX as a Web Component [p 215].

Selector parameters

The following parameters are only applicable to fields that return an enumeration, foreign key or `osd:resource`.

Parameter	Description
`selector`	Specifies whether: • `true`: returns the number of all possible values • `false`: returns the number of possible values for the current field. `Boolean` type, default value is `false`. **Note** This parameter is ignored with the `history` category.
`selectorFilter`	Specifies the filter of the selector. `String` type value, the syntax complies with the Recherche textuelle [p 126].

HTTP codes

HTTP code	Description
`200` *(OK)*	The selected resource is successfully counted.
`400` *(Bad request)*	The request is incorrect. This occurs when: • The selected field in a record or a dataset is sub-terminal. • The selected dataset field is a dataset tree. • The XPath predicate of the `filter` parameter is malformed or contains unfilterable nodes. • The table view for the `viewPublication` parameter is either hierarchical, non-existent or non-published. • The `selector` parameter is used for a non-enumerated node, or the `firstElementIndex` is negative, higher than or equal to the number of values.
`403` *(Forbidden)*	The selected resource is hidden for the authenticated user.
`404` *(Not found)*	The selected resource is not found.

Optimistic locking

To prevent an update or a delete operation on a record that was previously read but may have changed in the meantime, an optimistic locking mechanism is provided.

To enable optimistic locking, a select request must set the parameter `includeTechnicals` to `true`.

See Technical data [p 742] for more information.

The value of the `lastUpdateDate` property must be included in the following update request. If the record has been changed since the specified time, the update or delete will be cancelled.

- **Record**: update whole or partial content of the selected record.

  The property `lastUpdateDate` should be added to the request body to prevent update of a modified record.

  See the JSON [p 721] example of a record.

- **Field**: update of a single field of the selected record.

  The property value `lastUpdateDate` must be declared in the request URL by the `checkNotChangedSinceLastUpdateDate` parameter to prevent the update of a modified record.

The property value `lastUpdateDate` can also be used in the request URL `checkNotChangedSinceLastUpdateDate` parameter to prevent deletion on a modified record.

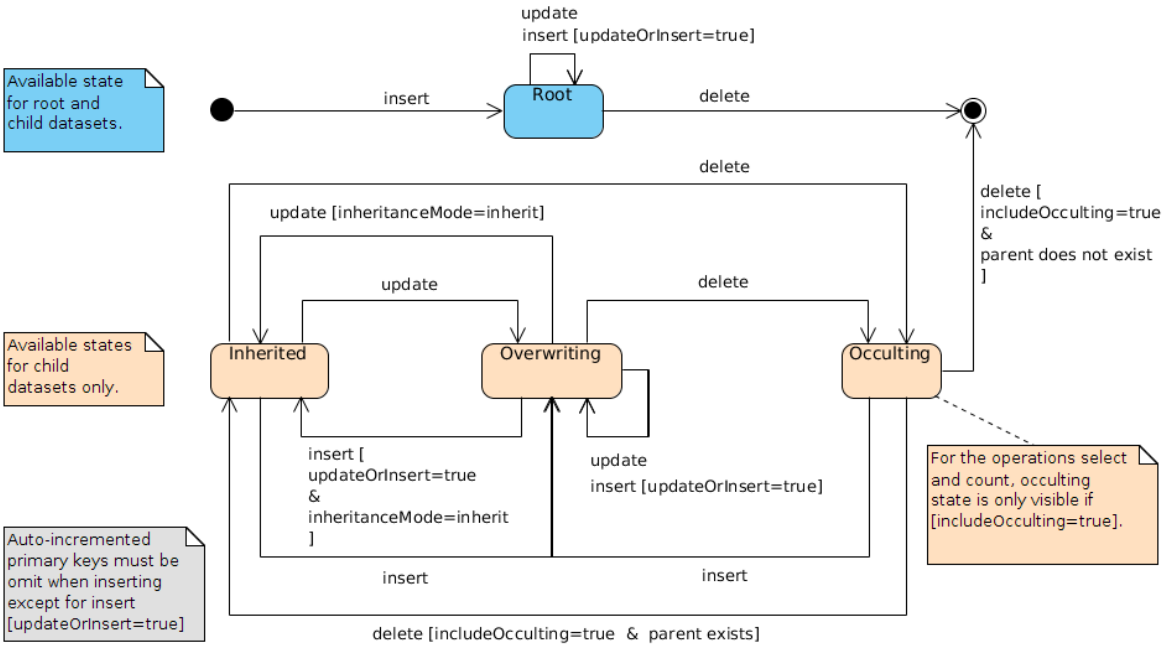
> **Note**
>
> The `checkNotChangedSinceLastUpdateDate` parameter may be used more than once but only on the same record. This implies that if the request URL returns more than one record, the request will fail.

Inheritance

EBX inheritance features are supported by built-in RESTful services using specific properties and automatic behaviors. In most cases, the inheritance state will be automatically computed by the server according to the record and field definition or content. Every action that modifies a record or a field may have an indirect impact on those states. In order to fully handle the inheritance life cycle, direct modifications of the state are allowed under certain conditions. Forbidden or incoherent explicit alteration attempts are ignored.

Voir aussi*Hritage et rsolution de valeur [p 294]*

Record inheritance life cycle in built-in RESTful services



Inheritance properties

The following table describes properties related to the EBX inheritance features.

Property	Location	Description
`inheritance`	record or table metadata	Specifies if dataset inheritance is activated for the table. The value is computed from the data model and cannot be modified through built-in RESTful services. **Voir aussi** *`inheritance` property in metadata.* [p 731]
`inheritedField`	field metadata	Specifies the field's value source. The source data are directly taken from the data model and cannot be modified through built-in RESTful services. **Voir aussi** *`inheritedField` property in metadata.* [p 732]
`inheritanceMode`	record in child dataset	Specifies the record's inheritance state. To set a record's inheritance from `overwrite` to `inherit`, its `inheritanceMode` value must be explicitly provided in the request. In this specific case, the `content` property will be ignored if present. `occult` and `root` explicit values are always ignored. An `overwrite` explicit value without a `content` property is ignored. **Note** Inherited record's fields are necessarily `inherit`. **Note** Root records in child dataset will always be `root`. Possible values are: `root`, `inherit`, `overwrite`, `occult`. For more information, see Mcanisme de rsolution d'enregistrement [p 296].
	field in `overwrite` record	Specifies the field's inheritance state. To set a field's inheritance to `inherit`, its `inheritanceMode` value must be explicitly provided in the request. The `content` property will be ignored in this case. `overwrite` explicit value without a `content` property is ignored. **Note** `inheritanceMode` at field level does not appear for `root`, `inherit` and `occult` records. **Note** `inheritedFieldMode` and `inheritanceMode` properties cannot be both set on the same field. Possible values are: `inherit`, `overwrite`. For more information, see Hritage et rsolution de valeur [p 294].

Property	Location	Description
`inheritedFieldMode`	inherited field	Specifies the inherited field's inheritance state. To set a field's inheritance to `inherit`, its `inheritedFieldMode` value must be explicitly provided in the request. The `content` property will be ignored in this case. `overwrite` explicit values without a `content` property are ignored. **Note** `inheritedFieldMode` and `inheritanceMode` properties cannot be both set on the same field. **Note** `inheritedFieldMode` has priority over the `inheritanceMode` property. Possible values are: `inherit`, `overwrite`. For more information, see Mcanisme de rsolution de valeur [p 297].

99.7 Limitations

General limitations

- Indexes, in the request URL `{pathInDataset}` or `{pathInRecord}`, are not supported.
- Nested aggregated lists are not supported.
- Dataset nodes and field operations applied to nodes that are sub-terminal are not supported.

  See Access properties [p 561] for more information about terminal nodes.

Read operations

- Within the `selector`, the pagination context is limited to the `nextPage` property.
- Within the `viewPublication` parameter, the hierarchical view is not supported.
- The `sortOnLabel` parameter ignores programmatic labels.
- The system information response's properties cannot be browsed through the REST URL with the hierarchical representation.

  See System information operation [p 689] for more information.

Write operations

- Association fields cannot be updated, therefore, the list of associated records cannot be modified directly.
- Control policy `onUserSubmit-checkModifiedValues` of the user interface is not supported. To retrieve validation errors, invoke the select operation on the resource by including the `includeValidation` parameter.

  See Blocking and non-blocking constraints [p 546] for more information.

Directory operations

- Changing or resetting a password for a user is not supported.

CHAPITRE **100**

JSON format

Ce chapitre contient les sections suivantes :

1. Introduction
2. Global structure
3. Meta-data
4. Sort criteria information
5. Validation
6. Content
7. Update modes
8. Known limitations

100.1 Introduction

The JSON (JavaScript Object Notation) is the data-interchange format used by TIBCO EBX RESTful operations [p 681].

This format is lightweight, self-describing and can be used to design UIs or to integrate EBX in a company's information system.

- The data context is exhaustive, except for association fields and selection nodes that are not directly returned. However, these fields are included in the response with a URL link named `details` included by default. It can be indirectly used to get the fields content.
- The volume of data is limited by a pagination mechanism activated by default, it can be configured or disabled.

URL formatted links allow retrieving:

- Tables, records, dataset non-terminal nodes, foreign keys, resource fields (property `details`).
- Possible values for foreign keys or enumerations (`selector` parameter).

Voir aussi*Activation and configuration [p 674]*

> **Note**
>
> JSON data are always encoded in UTF-8.

100.2 Global structure

JSON Request body

Request body is represented by a JSON `Object` whose content varies according to the operation and its category.

Auth category

The request body holds several properties directly placed in the root JSON `Object`.

- **Token creation**

  Specifies the `login` and `password` to use for an authentication token creation attempt.

```
{
  "login": "...",                 // JSON String
  "password": "..."               // JSON String
}
```

  Specifies the `specific` attribute, to activate the user authentication against the HTTP request, for an authentication token creation attempt.

```
{
  "specific": true               // JSON Boolean
}
```

  Voir aussi*Create token operation* *[p 690]*

- **Password change**

  Specifies the `login`, `password` and `passwordNew` to use for the password change.

```
{
  "login": "...",                 // JSON String
  "password": "...",              // JSON String
  "passwordNew": "..."            // JSON String
}
```

  Voir aussi*Change password operation* *[p 692]*

Data category

The request body contains at least a `content` property where master data values will be defined.

- **Dataset node**

  Specifies the target values of terminal nodes under the specified node. This request is used on the dataset node update operation.

```
{
   "content": {
      "nodeName1": {
         "content": true
      },
      "nodeName2": {
         "content": 2
      },
      "nodeName3": {
         "content": "Hello"
      }
   }
}
```

Voir aussi*Update operation* *[p 706]*

- **Record**

  Specifies the target record content by setting the value for each field. For missing fields, the behavior depends on the request parameter `byDelta`. This structure is used on table record insert or on record update.

  Voir aussi*Inheritance* *[p 714]*

  Some technical data can be added beside the `content` property such as `lastUpdateDate`.

  Voir aussi*Optimistic locking* *[p 713]*

```
{
   ...
   "lastUpdateDate": "2015-12-25T00:00:00.001",
   ...
   "content": {
      "gender": {
         "content": "Mr."
      },
      "lastName": {
         "content": "Chopin"
      },
      "lastName-en": {
         "content": "Chopin",
         "inheritedFieldMode": "inherit"
      },
      "firstName": {
         "content": "Fryderyk"
      },
      "firstName-en": {
         "content": "Frdric",
         "inheritedFieldMode": "overwrite"
      },
      "birthDate": {
         "content": "1810-03-01"
      },
      "deathDate": {
         "content": "1849-10-17"
      },
      "jobs": {
         "content": [
            {
               "content": "CM"
            },
            {
               "content": "PI"
            }
         ]
      },
      "infos": {
         "content": [
            {
               "content": "https://en.wikipedia.org/wiki/Chopin"
            }
         ]
      }
   }
}
```

  Voir aussi

  Insert operation *[p 701]*

  Update operation *[p 706]*

- **Record fields**

Specifies the target values of fields under the record terminal node by setting the value of each field. For missing fields, the behavior depends on the request parameter `byDelta`. This structure is only used for table record updates.

Voir aussi*Inheritance* *[p 714]*

```
{
   "content": [
      {
         "content": "CM"
      },
      {
         "content": "PI"
      }
   ]
}
```

Voir aussi*Update operation* *[p 706]*

- **Record table**

  Defines the content of one or more records by indicating the value of each field. For missing fields, the behavior depends on the parameter of the request `byDelta`. This structure is used upon insert or update of records in the table.

```
{
   "rows": [
      {
         "content": {
            "gender": {
               "content": "M"
            },
            "lastName": {
               "content": "Saint-Sans"
            },
            "firstName": {
               "content": "Camille"
            },
            "birthDate": {
               "content": "1835-10-09"
            },
            ...
         }
      },
      {
         "content": {
            "gender": {
               "content": "M"
            },
            "lastName": {
               "content": "Debussy"
            },
            "firstName": {
               "content": "Claude"
            },
            "birthDate": {
               "content": "1862-10-22"
            },
            ...
         }
      }
   ]
}
```

  Voir aussi

  Insert operation *[p 701]*

  Update operation *[p 706]*

- **Record table to be deleted**

Defines one or more records. This structure is used upon deleting several records from the same table.

```
{
   "rows": [
      {
         "details": "http://.../root/table/1"
      },
      {
         "details": "http://.../root/table/2"
      },
      {
         "primaryKey": "./oid=3"
      },
      {
         "foreignKey": "4"
      },
      ...
   ]
}
```

Voir aussi*Delete operation* *[p 708]*

- **Field**

  Specifies the target field content. This request is used on the field update.

  The request has the same structure as defined in node value [p 735] by only keeping the `content` entry. Other entries are simply ignored.

  Voir aussi*Update operation* *[p 706]*

- **Open or close user interface**

  Specifies whether the user interface is open or close and the unavailability message.

```
{
    "content": {
        "toolStatus": {
            "content": true // or false
        },
        "toolStatusCloseMessage": {
            "content": "Access is temporarily forbidden for maintenance."
        }
    }
}
```

  Voir aussi*User interface operations* *[p 689]*

Only writable fields can be mentioned in the request, this excludes the following cases:

- Association node,
- Selection node,
- Value function,
- JavaBean field that does not have a setter,
- Unwritable permission on node for authenticated user.

JSON Response body

The response body is represented by a JSON `Object` whose content depends on the operation and its category.

Admin category

The selection operation for this category only provides the values corresponding to the request under a `content` property.

- **System information**

  Contains EBX instance's system information. The representation of these data can be flat or hierarchical.

  Flat representation:

```
{
 "content": {
  "bootInfoEBX": {
   "label": "EBX® configuration",
   "content": {
    "product.version": {
     "label": "EBX® product version",
     "content": "5.8.1 [...] Enterprise Edition"
    },
    "product.configuration.file": {
     "label": "EBX® main configuration file",
     "content": "System property [ebx.properties=./ebx.properties]"
    }
    // others keys
   }
  },
  "repositoryInfo": {
   "label": "Repository information",
   "content": {
    "repository.identity": {
     "label": "Repository identity",
     "content": "00905A5753FD"
    },
    "repository.label": {
     "label": "Repository label",
     "content": "My repository"
    }
    // others keys
   }
  },
  "bootInfoVM": {
   "label": "System information",
   "content": {
    "java.home": {
     "label": "Java installation directory",
     "content": "C:\\JTools\\jdk1.8.0\\jre"
    },
    "java.vendor": {
     "label": "Java vendor",
     "content": "Oracle Corporation"
    }
    // others keys
   }
  }
 }
}
```

  Hierarchical representation:

```
{
 "content": {
  "bootInfoEBX": {
      "label": "EBX® configuration",
   "content": {
    "product": {
     "content": {
      "version": {
       "label": "EBX® product version",
       "content": "5.8.1 [...] Enterprise Edition"
      },
      "configuration": {
       "content": {
        "file": {
         "label": "EBX® main configuration file",
         "content": "System property [ebx.properties=./ebx.properties]"
        }
       }
      }
```

```
     }
    },
    "vm": {
     "content": {
      "startTime": {
       "label": "VM start time",
       "content": "2017/09/11-10:04:17-0729 CEST"
      },
      "identifier": {
       "label": "VM identifier",
       "content": "1"
      }
     }
    }
    // other hierarchical keys
   }
  }
 }
}
```

Voir aussi*System information operation* *[p 689]*

Auth category

The response body contains several properties directly placed in its root JSON `object`.

- **Token creation**

  Contains the token value and its type.

```
{
  "accessToken": "...",            // JSON String
  "tokenType": "..."               // JSON String
}
```

Voir aussi*Create token operation* *[p 690]*

Data category

The selection operation contains two different parts.

The first one named `meta` contains the exhaustive structure of the response.

The other, regrouping `content`, `rows`, `pagination`...etc, contains the values corresponding to the request.

- **Dataset tree**

  Contains the hierarchy of `table` and non-terminal `group` nodes.

```
{
   "meta": {
      "fields": [
         {
            "name": "rootName",
            "label": "Localized label",
            "description": "Localized description",
            "type": "group",
            "pathInDataset": "/rootName",
            "fields": [
               {
                  "name": "settings",
                  "label": "Settings",
                  "type": "group",
                  "pathInDataset": "/rootName/settings",
                  "fields": [
                     {
                        "name": "settingA",
                        "label": "A settings label",
                        "type": "group",
                        "pathInDataset": "/rootName/settings/settingA"
                     },
                     {
                        "name": "settingB",
```

```
                        "label": "B settings label",
                        "type": "group",
                        "pathInDataset": "/rootName/settings/settingB"
                     }
                  ]
               },
               {
                  "name": "table1",
                  "label": "Table1 localized label",
                  "type": "table",
                  "minOccurs": 0,
                  "maxOccurs": "unbounded",
                  "pathInDataset": "/rootName/table1"
               },
               {
                  "name": "table2",
                  "label": "Table2 localized label",
                  "type": "table",
                  "minOccurs": 0,
                  "maxOccurs": "unbounded",
                  "pathInDataset": "/rootName/table2"
               }
            ]
         }
      ]
   },
   "validation": [
      {
         "level": "error",
         "message": "Value must be greater than or equal to 0.",
         "details": "http://.../rootName/settings/settingA/settingA1?includeValidation=true"
      },
      {
         "level": "error",
         "message": "Field 'Settings A2' is mandatory.",
         "details": "http://.../rootName/settings/settingA/settingA2?includeValidation=true"
      }
   ],
   "content": {
      "rootName": {
         "details": "http://.../rootName",
         "content": {
            "settings": {
               "details": "http://.../rootName/settings",
               "content": {
                  "weekTimeSheet": {
                     "details": "http://.../rootName/settings/settingA"
                  },
                  "vacationRequest": {
                     "details": "http://.../rootName/settings/settingB"
                  }
               }
            },
            "table1": {
               "details": "http://.../rootName/table1"
            },
            "table2": {
               "details": "http://.../rootName/table2"
            }
         }
      }
   }
}
```


The `meta` and `validation` properties are optional.

Voir aussi

Meta-data [p 730]

Validation [p 734]

Select operation [p 693]

- **Dataset node**

  Contains the list of terminal nodes under the specified node.


```
{
   "meta": {
      "fields": [
         {
```

```
            "name": "nodeName1",
            "label": "Localized label of the field node 1",
            "description": "Localized description",
            "type": "boolean",
            "minOccurs": 1,
            "maxOccurs": 1,
            "pathInDataset": "/rootName/.../nodeName1"
         },
         {
            "name": "nodeName2",
            "label": "Localized label of the field node 2",
            "type": "int",
            "minOccurs": 1,
            "maxOccurs": 1,
            "pathInDataset": "/rootName/.../nodeName2"
         }
      ]
   },
   "content": {
      "nodeName1": {
         "content": true
      },
      "nodeName2": {
         "content": -5,
         "validation": [
            {
               "level": "error",
               "message": "Value must be greater than or equal to 0."
            }
         ]
      }
   }
}
```

Voir aussi*Select operation* *[p 693]*

- **Table**

  JSON `Object` containing the properties:

  - (Optional) The table meta data [p 730],
  - (Optional) The sort criteria applied,
  - (Optional) The table validation report,
  - `rows` containing a table of selected records. Each record is represented by an object, if no record is selected then the table is empty.
  - (Optional) `pagination` containing pagination [p 739] information if activated.

```
{
   "rows": [
      {
         "label": "Claude Levi-Strauss",
         "details": "http://.../root/individu/1",
         "content": {
            "id": {
               "content": 1
            },
         ...
         }
      },
      {
         "label": "Sigmoud Freud",
         "details": "http://.../root/individu/5",
         "content": {
            "id": {
               "content": 2
            },
         ...
         }
      },
   ...
      {
         "label": "Alfred Dreyfus",
         "details": "http://.../root/individu/10",
         "content": {
            "id": {
```

```
               "content": 30
            },
   ...
         }
      }
   ],
   "sortCriteria": [
      {
         "path": "/name",
         "order": "lasc"
      },
      ...
   ],
   "pagination": {
      "firstPage": null,
      "previousPage": null,
      "nextPage": "http://.../root/individu?pageRecordFilter=./id=9&pageSize=9&pageAction=next",
      "lastPage": "http://.../root/individu?pageSize=9&pageAction=last"
   }
}
```

Voir aussi*Select operation* *[p 693]*

- **Record**

  JSON `object` containing:

  - The label,
  - (Optional) The record URL,
  - (Optional) The technical data [p 742],
  - (Optional) The table metadata [p 730],
  - (Optional) The record validation report,
  - (Optional) The inheritance mode of the record is: `root`, `inherit`, `overwrite` or `occult`. This value is available for a child dataset,

    Voir aussi

    Mcanisme de rsolution d'enregistrement *[p 296]*

    Inheritance *[p 714]*
  - The record content.

```
{
   "label": "Name1",
   "details": "http://.../rootName/table1/pk1",
   "creationDate": "2015-02-02T19:00:53.142",
   "creationUser": "admin",
   "lastUpdateDate": "2015-09-01T17:22:24.684",
   "lastUpdateUser": "admin",
   "inheritanceMode": "root",
   "meta": {
      "name": "table1",
      "label": "Table1 localized label",
      "type": "table",
      "minOccurs": 0,
      "maxOccurs": "unbounded",
      "primaryKeys": [
         "/pk"
      ],
      "inheritance": "true",
      "fields": [
         {
            "name": "pk",
            "label": "Identifier",
            "type": "string",
            "minOccurs": 1,
            "maxOccurs": 1,
            "pathInRecord": "pk",
            "filterable": true,
            "sortable": true
         },
```

```
         {
            "name": "name",
            "label": "Name",
            "type": "string",
            "minOccurs": 1,
            "maxOccurs": 1,
            "pathInRecord": "name",
            "filterable": true,
            "sortable": true
         },
         {
            "name": "name-fr",
            "label": "Nom",
            "type": "string",
            "minOccurs": 1,
            "maxOccurs": 1,
            "inheritedField": {
                "sourceNode": "./name"
            },
            "pathInRecord": "name-fr",
            "filterable": true,
            "sortable": true
         },
         {
            "name": "parent",
            "label": "Parent",
            "description": "Localized description.",
            "type": "foreignKey",
            "minOccurs": 1,
            "maxOccurs": 1,
            "foreignKey": {
               "tablePath": "/rootName/table1",
               "details": "http://.../rootName/table1"
            },
            "enumeration": "foreignKey",
            "pathInRecord": "parent",
            "filterable": true,
            "sortable": true
         }
      ]
   },
   "content": {
      "pk": {
         "content": "pk1"
      },
      "name": {
         "content": "Name1"
      },
      "name-fr": {
         "content": "Name1",
         "inheritedFieldMode": "inherit"
      },
      "parent": {
         "content": null,
         "validation":
         [
            {
               "level": "error",
               "message": "Field 'Parent' is mandatory."
            }
         ]
      }
   },
   "validation": {
      ...
   }
}
```


Voir aussi*Select operation* *[p 693]*

- **Fields**

  For association or selection nodes, contains the target table with associated records if, and only if, the `includeDetails` parameter is set to `true`.

  For other kinds of nodes, contains the current node value [p 735], by adding `meta` entry if enabled.

  Voir aussi*Select operation* *[p 693]*

- **Retrieve the user interface state**

  Contains the user interface status and the unavailability message.

```
{
    "content": {
        "toolStatus": {
            "content": true,
            "label": "Open",
            "selector": "http://.../domain/toolStatus/toolStatus?selector=true",
        },
        "toolStatusCloseMessage": {
            "content": "Access is temporarily forbidden for maintenance.",
        }
    }
}
```

Voir aussi*User interface operations* *[p 689]*

> **Note**
>
> Node, records and field in `meta`, `rows` and `content` may be hidden depending on their resolved permissions *(see permissions [p 299])*.

100.3 Meta-data

This section can be activated on demand with the `includeMeta` [p 696] parameter. It describes the structure and the JSON typing of the `content` section.

This section is deactivated by default for selection operations.

Voir aussi*Select operation* *[p 693]*

Structure of table

Table meta-data is represented by a JSON `object` with the following properties:

JSON property	JSON format	Description	Required
`name`	`String`	Name of the table defined in schema.	Yes
`label`	`String`	Table label. If undefined, the name of the schema node is returned.	Yes
`description`	`String`	Table description.	No
`type`	`String`	Always equal to: `table`.	Yes
`minOccurs`	`Number`	Number of minimum authorized record(s).	Yes
`maxOccurs`	`Number` or `String`	Number of maximum authorized record(s) or `unbounded`.	Yes
`history`	`Boolean`	Specifies if the table content is historized. Its value is `true` if history is activated, `false` otherwise. **Voir aussi***Historique* *[p 273]*	No
`primaryKeyFields`	`Array`	Array of the paths corresponding to the primary key.	Yes
`inheritance`	`Boolean`	Specifies whether the dataset inheritance is activated for the table. Its value is `true` if inheritance is activated, `false` otherwise. **Voir aussi***Hritage et rsolution de valeur* *[p 294]*	No
`fields`	`Array`	Array of fields, that are direct children of the record node. Each field may also recursively contain sub-fields.	Yes

Structure of field

Each authorized field is represented by a JSON `object` with the following properties:

JSON property	JSON format	Description	Required
`name`	`String`	Name of the current authorized schema node.	Yes
`label`	`String`	Node label. If undefined, the name of the schema node is returned.	Yes
`description`	`String`	Node description.	No
`type`	`String`	Node type: simple type [p 737], `group`, `table`, `foreignKey`, etc.	Yes
`minOccurs`	`Number`	Number of minimum authorized occurrence(s).	Yes
`maxOccurs`	`Number` or `String`	Number of maximum authorized occurrence(s) or `unbounded`.	Yes
`inheritedField`	`Object`	Holds information related to the inherited field's value source. `"inheritedField": { "sourceRecord": "/path/to/record", // (optional) "sourceNode": "./path/to/Node" }` **Voir aussi***Hritage et rsolution de valeur [p 294]*	No
`foreignKey`	`Object`	Contains information related to the target table. `{ "dataspace": "BAuthors", "dataset": "Authors", "tablePath": "/root/Authors", "details": "http://.../BAuthors/Authors/root/Authors" }`	No (*)
`dataspace`	`String`	Target dataspace or snapshot identifier. This property is placed under the `foreignKey` property.	No (*)
`dataset`	`String`	Target dataset identifier. This property is placed under the `foreignKey` property.	No (*)
`tablePath`	`String`	Target table path. This property is placed under the `foreignKey` property.	Yes
`details`	`String`	Target table URL. This property is placed under the `foreignKey` property and is included by default.	No
`enumeration`	`String`	Specifies if the field is an enumeration value. Possible values are: • `foreignKey`	No

JSON property	JSON format	Description	Required
		• `static` • `dynamic` • `programmatic` • `nomenclature` • `resource` **Voir aussi** *`SchemaFacetEnumeration`*[API] Specifies whether the field can be used for retrieving possible values by using the `selector` request parameter.	
`valueFunction`	`Boolean`	Specifies if the field is a computed value. **Voir aussi** *Computed values* [p 551]	No
`pathInDataset`	`String`	Relative field path starting from the schema node.	No (**)
`pathInRecord`	`String`	Relative field path starting from the table node.	No (*)
`filterable`	`Boolean`	Specifies whether the field can be used for filtering record using `filter` request parameter.	No (*)
`sortable`	`Boolean`	Specifies whether the field can be used in sort criteria using `sort` request parameter.	No (*)
`fields`	`Array` of `Object` elements	Contains the structure and typing of each group field.	No

(*) Only available for table, record and record field operations.

(**) Only available for dataset tree operations.

100.4 Sort criteria information

The sort criteria applied to the request can be returned on demand, by using the `includeSortCriteria` [p 697] parameter (deactivated by default). If it is activated, a `sortCriteria` property is directly added to the response root node.

A `sortCriteria` property contains a JSON `Array` that contains ordered sort criteria, and for each sort criterion, a JSON `Object` is added with the following properties:

JSON property	JSON format	Description	Required
`path`	`String`	Field path.	Yes
`order`	`String`	Possible values are: `asc`, `lasc`, `desc` or `ldesc`.	Yes

100.5 **Validation**

The validation can be activated on demand with the `includeValidation` [p 697] parameter (deactivated by default). If it is activated, `validation` properties are directly added on target nodes with one or several messages. For messages without a target node path, a `validation` property is added on the root node.

A `validation` property contains a JSON `Array` and for each message, corresponding to a validation item, a JSON `Object` with properties:

JSON property	JSON format	Description	Required
`level`	`String`	Severity of the validation item, the possible values are: `info`, `warn`, `error`.	Yes
`message`	`String`	Description of the validation item.	Yes
`details`	`String` corresponding to an absolute URL.	URL of the resource associated with the validation item. Only available on the table and dataset scopes, if associated resources exist and if it is included. **Voir aussi** *`includeDetails` parameter* [p 696]	No

100.6 **Content**

This section can be deactivated on demand with the `includeContent` [p 696] parameter (activated by default). It provides the content of the record values, dataset, or field of one of the content fields for an authorized user. It also has additional information, including labels, technical information, URLs...

The content is represented by a JSON `Object` with a property set for each sub-node.

Node value

JSON property	JSON format	Description	Required
`content`	Content of simple type [p 737] Content of group and list [p 739]	Contains the node value. Available for all nodes except `association` and `selection`. However, their content can be retrieved by invoking the URL provided in `details`.	`No`
`details`	`String` corresponding to an absolute URL.	By invocation, the node details are returned. Response type after invocation depending on the meta type. • `foreignKey`: target record (available on table, record and field operation). • `resource`: target resource [p 542] (available on dataset node, table, record and field operations). • `association`: target table containing associated records (available on table and record operations). • `selection`: target table containing associated records (available on table and record operations). • `group`: target dataset group node (available on dataset tree operation). • `table`: target table (available on dataset tree operation). Example: `http://.../BReference/dataset/root/table/pk/associationField`	`No`
`label`	`String`	Contains the foreign key or enumeration label in the current locale. The default label is returned if the current locale is not supported.	`No`
`inheritanceMode`	`String`	Contains the node's inheritance state, considering only dataset inheritance. `inheritedFieldMode` and `inheritanceMode` properties cannot be both defined on the same node. **Voir aussi** *`inheritedFieldMode` [p 735]* *Mcanisme de rsolution d'enregistrement [p 296]*	`No`
`inheritedFieldMode`	`String`	Contains the node's field inheritance state, considering dataset and field inheritance. When both inheritances are used, field inheritance has priority over the dataset one. `inheritedFieldMode` and `inheritanceMode` properties cannot be both defined on the same node. **Voir aussi** *`inheritanceMode` [p 735]* *Mcanisme de rsolution de valeur [p 297]*	`No`
`selector`	`String`	Correspond to the URL for selector [p 740] operation.	`No`

JSON property	JSON format	Description	Required
	corresponding to an absolute URL.	Example: `http://.../BReference/dataset/root/table/pk/enumField?selector=true`	
`validation`	`Array`	Contains the validation report that concerns the current node context.	`No`

Content of simple type

A simple field value is stored in a JSON object and the content is the value of the `content` property.

XML Schema	JSON format	Examples	Meta type

`xs:string`
`xs:Name`
`osd:html`
`osd:email`
`osd:text`
	`String` (Unicode characters, cf. RFC4627)	"A text"
		"The escape of \"special character\" is preceded by a backslash."
		" An HTML tag can thus be written without trouble "
		"employee@mycompany.com"
		null
			`string`
			`name`
			`html`
			`email`
			`text`

`osd:locale`	`String` (Language tag, cf. RFC1766)	"en-US"	`locale`
`xs:string` (Foreign key)	`String` contains the value of the formatted foreign key.	`"0"` `"true|99"`	`foreignKey`
`xs:boolean`	`Boolean`	true false null	`boolean`
`xs:decimal`	`Number` or `null`	-10.5 20.001 15 -1e-13	`decimal`
`xs:date`	`String` with format: "yyyy-MM-dd"	"2015-04-13"	`date`
`xs:time`	`String` with format: • "HH:mm:ss" • "HH:mm:ss.SSS"	"11:55:00" "11:55:00.000"	`time`
`xs:dateTime`	`String` with format: • "yyyy-MM-ddTHH:mm:ss" • "yyyy-MM-ddTHH:mm:ss.SSS"	"2015-04-13T11:55:00" "2015-04-13T11:55:00.000"	`dateTime`
`xs:anyURI`	`String` (Uniform Resource Identifier, cf. RFC3986)	"https://fr.wikipedia.org/wiki/Ren_Descartes"	`anyURI`
`xs:int` `xs:integer`	`Number` or `null`	1596	`int`
`osd:resource`	`String`	"ebx-tutorial:ext-images:frontpages/Ajax for Dummies.jpg"	`resource`

XML Schema	JSON format	Examples	Meta type
	contains the resource formatted value.		
`osd:color`	`String` with format: "#[A-Fa-f0-9]{6}" contains the formatted value for the color.	"#F6E0E0"	`color`
`osd:datasetName`	`String` with format: "[a-zA-Z_][-a-zA-Z0-9_.]*" and 64 characters max. contains the formatted value of the dataset name.	"ebx-tutorial"	`dataset`
`osd:dataspaceKey`	`String` with format: "[BV][a-zA-Z0-9_:.\\-\\]*" and 33 characters max. contains the formatted key value of the dataspace.	"Bebx-tutorial"	`dataspace`

Content of group and list

XML Schema	JSON format	Examples	Meta type
Group `xs:complexType`	`Object` Contains a property per sub-node.	Example for a simple-occurrence group. `{ "road" : {"content" : "11 rue scribe"}, "zipcode" : {"content" : "75009"}, "country" : {"content" : "France"} }`	`group`

List
`maxOccurs > 1`
	`Array`
	Contains an array of all field occurrences represented by a JSON `Object`.
	Each object is represented as a node value [p 735].
		Example for a multi-occurrence field of the `xs:int` type.
		`[ {"content": 0}, {"content": 1}, {"content": 2}, {"content": 3} ]`
		Example for a multi-occurrence group.
		`[ { "content": { "road": {"content": "11 rue scribe"}, "zipcode": {"content": "75009"}, "country": {"content": "France"} } }, { "content": { "road": {"content": "711 Atlantic Ave"}, "zipcode": {"content": "MA 02111"}, "country": {"content": "United States"} } } ]`
			Meta of simple type [p 737],
			or
			`group`

Pagination

This feature allows returning a limited and parameterizable number of data. Pagination can be applied to data of the following types: records, association values, selection node values and selectors. A context named `pagination` is returned only if it has been activated. This context allows browsing data similarly to the UI.

Pagination is activated by default.

Voir aussi*Select operation [p 693]*

Detailed information related to this context can be found hereafter:

JSON property	JSON format	Description	Required
`firstPage`	`String` or `null` (*)	URL to access the first page.	Yes (**)
`previousPage`	`String` or `null` (*)	URL to access the previous page.	Yes (**)
`nextPage`	`String` or `null` (*)	URL to access the next page.	Yes
`lastPage`	`String` or `null` (*)	URL to access the last page.	Yes (**)

> **Note**
>
> (*) Only defines if data is available in this context and not in the response.

> **Note**
>
> (**) Not present on selector.

Selector

By invoking the URL represented by the property `selector` on a field that provides an enumeration, this returns a JSON `Object` containing the properties:

- `rows` containing an `Array` of JSON `Object` where each one contains two entries, such as the returned `content` that can be persisted and the corresponding `label`. The list of possible items is established depending on the current context.
- (Optional) `pagination` containing pagination [p 739] information (activated by default).

```
{
   "rows": [
      {
         "content": "F",
         "label": "feminine"
      },
      {
         "content": "M",
         "label": "masculine"
      }
   ],
   "pagination": {
      "nextPage": null
   }
}
```

Voir aussi*includeSelector [p 697]*

Insert operation report

When invoking the insert operation with a record table, it can optionally return a report. The report includes a JSON `Object` that contains the following properties:

- `rows` contains a JSON `ObjectArray`, where each element corresponds to the result of a request element.
- `code` contains an `int` of the JSON `Number` type, and allows to know whether the record has been inserted or updated. This property is included if, and only if, the `updateOrInsert` parameter is set to `true`.

- `foreignKey` contains a `string` of the JSON `String` type, corresponding to the content to be used as a foreign key for this record. This property is included if, and only if, the parameter `includeForeignKey` is set to `true`.
- `label` contains a `string` of the JSON `String` type, and allows to retrieve the record label. This property is included if, and only if, the parameter `includeLabel` is set to `yes`.
- `details` containing a `string` of the JSON `String` type, corresponding to the resource URL. This property is included if, and only if, the parameter `includeDetails` is set to `true`.

```
{
   "rows": [
      {
         "code": 204,
         "foreignKey": "62",
         "label": "Claude Debussy",
         "details": "http://.../root/individu/62"
      },
      {
         "code": 201,
         "foreignKey": "195",
         "label": "Camille Saint-Sans",
         "details": "http://.../root/individu/195"
      }
   ]
}
```

Voir aussi*Insert operation* *[p 701]*

Delete operation report

When invoking the delete operation, a report is returned. The report includes a JSON `Object` that contains the following properties:

- `deletedCount` containing an integer of the JSON `Number` type, corresponds to the number of deleted records.
- `occultedCount` containing an integer of the JSON `Number` type, corresponds to the number of occulted records.
- `inheritedCount` containing an integer of the JSON `Number` type, corresponds to the number of inherited records.

```
{
 "deletedCount": 1,
 "occultedCount": 0,
 "inheritedCount": 0
}
```

Voir aussi*Delete operation* *[p 708]*

Technical data

Each returned record is completed with the properties corresponding to its technical data, containing:

JSON property	JSON format	Description	Required
`creationDate`	`String`	Creation date.	Yes
`creationUser`	`String`	Creation user identifier.	Yes
`lastUpdateDate`	`String`	Last update date.	Yes
`lastUpdateUser`	`String`	Last update user identifier.	Yes

```
{
   ...
   "creationDate": "2015-12-24T19:00:53.158",
   "creationUser": "admin",
   "lastUpdateDate": "2015-12-25T00:00:00.001",
   "lastUpdateUser": "admin",
   ...
}
```

100.7 **Update modes**

The `byDelta` mode allows to ignore data model elements that are missing from the JSON source document. This mode is enabled (by default) through RESTful operations. The following table summarizes the behavior of insert and update operations when elements are not included in the source document.

See the RESTful data services operations update [p 706] and insert [p 701], as well as `ImportSpec.setByDelta`[API] in the Java API for more information.

State in the JSON source document	Behavior
The property does not exist in the source document	**If the `byDelta` mode is activated (default):** • For the `update` operation, the field value remains unchanged. • For the `insert` operation, the behavior is the same as when the `byDelta` mode is disabled. **If the `byDelta` mode is disabled through the RESTful operation parameter:** The target field is set to one of the following values: • If the element defines a default value, the target field is set to that default value. • If the element is of a type other than a string or list, the target field value is set to `null`. • If the element is an aggregated list, the target field value is set to an empty list value. • If the element is a string that differentiates `null` from an empty string, the target field value is set to `null`. If it is a string that does not differentiate the two, an empty string. • If the element (simple or complex) is hidden in the data services, the target value remains unchanged. Voir aussi *Hiding a field in Data Services* [p 564] **Note** The user performing the import must have the required permissions to create or change the target field value. Otherwise, the operation will be aborted.
The element is present and its value is `null` (for example, `"content": null`)	The target field is always set to `null` except for lists, in which case it is not supported.

100.8 Known limitations

Field values

The value of fields `xs:date`, `xs:time` and `xs:dateTime` does not contain a time zone associated with the JSON-primitive type.

CHAPITRE 101

REST Toolkit

Ce chapitre contient les sections suivantes :

1. Introduction
2. Application definitions
3. Service and operation definitions
4. Authentication and lookup mechanism
5. REST authentication and permissions
6. URI builders
7. Exception handling
8. Monitoring
9. Packaging and registration

101.1 Introduction

TIBCO EBX offers the possibility to develop custom REST services using the REST Toolkit. The REST Toolkit supports JAX-RS 2.1 (JSR-370) and JSON-B (JSR-367).

A REST service is implemented by a Java class and its operations are implemented by Java methods. The response can be generated by serializing POJO objects. The request input can be unserialized to POJOs. Various input and output formats, including JSON, are supported. For more details on supported formats, see media types [p 747].

Rest Toolkit supports the following:

- Injectable objects

  EBX provides injectable objects useful to authenticate the request's user, to access the EBX repository or to built URIs without worrying about the configuration (for example reverse-proxy [p 674] or REST forward [p 602] modes);

  JAX-RS injectable objects are also supported.

- Annotations

  EBX provides annotations to describe resources, grant anonymous access or add restrictions to a method.

  JAX-RS ans JSON-B annotations are also supported.

- logging and debugging utilities.

Voir aussi *JAX-RS: JavaTM API for RESTful Web Services 2.1*

101.2 Application definitions

An EBX module, that includes custom REST services, must provide at least one REST Toolkit application class. A REST Toolkit application class extends the EBX `RESTApplicationAbstract`[API] class. The minimum requirement is to define the base URL, using the `@ApplicationPath` annotation and the set of packages to scan for REST service classes.

> **Note**
>
> Only packages accessible from the web application's classloader can be scanned.

> **Note**
>
> It is possible to register REST resource classes or singletons, packaged inside or outside the web application archive, through the **ApplicationConfigurator.register(java.lang.Class)** `ApplicationConfigurator.register`[API] or **ApplicationConfigurator.register(java.lang.Object)** `ApplicationConfigurator.register`[API] methods.

> **Note**
>
> If no packages scope is defined, then every class reachable from the web application's classloader will be scanned.

The application path cannot be "`/`" and must not collide with an existing resource from the module. It is recommended to use "`/rest`" (the value of the `RESTApplicationAbstract.REST_DEFAULT_APPLICATION_PATH` constant).

EBX `Documentation`[API] annotation is optional. It is displayed to administrators in 'Technical configuration' > 'Modules and data models' or when logging and debugging.

```
import javax.ws.rs.*;

import com.orchestranetworks.rest.*;
import com.orchestranetworks.rest.annotation.*;

@ApplicationPath(RESTApplicationAbstract.REST_DEFAULT_APPLICATION_PATH)
@Documentation("My REST sample application")
public final class RESTApplication extends RESTApplicationAbstract
{
   public RESTApplication()
   {
      // Adds one or more package names which will be used to scan for components.
      super((cfg) -> cfg.addPackages(RESTApplication.class.getPackage()));
   }
}
```

101.3 Service and operation definitions

A REST Toolkit service is implemented by a Java class and its operations are implemented by its methods.

Class and methods can be annotated by `@Path` to specify the access path. The `@Path` annotation value defined at the class level will prepend the ones defined on methods. Warning, only one `@Path` annotation is allowed per class or method.

Media types accepted and produced by a resource are respectively defined by the `@Consumes` and `@Produces` JAX-RS annotations. The supported media types are:

- `application/json` (MediaType.APPLICATION_JSON_TYPE)
- `application/octet-stream` (MediaType.APPLICATION_OCTET_STREAM_TYPE)
- `application/x-www-form-urlencoded` (MediaType.APPLICATION_FORM_URLENCODED_TYPE)
- `multipart/form-data` (MediaType.MULTIPART_FORM_DATA_TYPE)
- `text/css`
- `text/html` (MediaType.TEXT_HTML_TYPE)
- `text/plain` (MediaType.TEXT_PLAIN_TYPE)

Valid HTTP(S) methods are specified by JAX-RS annotations `@GET`, `@POST`, `@PUT`, etc. Only one of these annotations can be set on each Java method (this means that a Java method can support only one HTTP method).

Warning: URL parameters with a name prefixed with `ebx-` are reserved by REST Toolkit and should not be defined by custom REST services, unless explicitly authorized by the EBX documentation.

URL and sample

The REST URL to access the description service for the sample is defined below:

`http[s]://<host>[:<port>]/<path to webapp>/rest/track/v1/description`

Where:

- `<path to webapp>` corresponds to the web application's path holding the REST Toolkit application, itself serving the sample service. The path is composed by multiple, or none, URI segments followed by the web application's name.

> **Note**
>
> Please note that `/rest/track/v1/description` corresponds to the concatenation of the application's `@ApplicationPath` and service's `@Path` annotations.

The following REST Toolkit service sample provides methods to query and manage track data:

```
import java.net.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;
import java.util.stream.*;

import javax.servlet.http.*;
import javax.ws.rs.*;
import javax.ws.rs.container.*;
import javax.ws.rs.core.*;

import com.orchestranetworks.rest.annotation.*;
import com.orchestranetworks.rest.inject.*;

/**
 * The REST Toolkit Track service v1.
 */
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
@Path("/track/v1")
@Documentation("Track service")
public final class TrackService
{
 @Context
 private ResourceInfo resourceInfo;
```

```
@Context
private SessionContext sessionContext;

private static final Map<Integer, TrackDTO> TRACKS = new ConcurrentHashMap<>();

/**
 * Gets service description
 */
@GET
@Path("/description")
@Documentation("Gets service description")
@Produces({ MediaType.TEXT_PLAIN, MediaType.APPLICATION_JSON })
@AnonymousAccessEnabled
public String handleServiceDescription()
{
 return this.resourceInfo.getResourceMethod().getAnnotation(Documentation.class).value();
}

/**
 * Selects tracks.
 */
@GET
@Path("/tracks")
@Documentation("Selects tracks")
public Collection<TrackDTO> handleSelectTracks(
  @QueryParam("singerFilter") final String singerFilter, // a URL parameter holding a Java regular expression
  @QueryParam("titleFilter") final String titleFilter) // a URL parameter holding a Java regular expression
{
 final Pattern singerPattern = TrackService.compilePattern(singerFilter);
 final Pattern titlePattern = TrackService.compilePattern(titleFilter);

 return TRACKS.values()
   .parallelStream()
   .filter(Objects::nonNull)
   .filter(track -> singerPattern == null || singerPattern.matcher(track.singer).matches())
   .filter(track -> titlePattern == null || titlePattern.matcher(track.title).matches())
   .collect(Collectors.toList());
}

private static Pattern compilePattern(final String aPattern)
{
 if (aPattern == null || aPattern.isEmpty())
  return null;

 try
 {
  return Pattern.compile(aPattern);
 }
 catch (final PatternSyntaxException ignore)
 {
  // ignore invalid pattern
  return null;
 }
}

/**
 * Counts all tracks.
 */
@GET
@Path("/tracks:count")
@Documentation("Counts all tracks")
public int handleCountTracks()
{
 return TRACKS.size();
}

/**
 * Selects a track by id.
 */
@GET
@Path("/tracks/{id}")
@Documentation("Selects a track by id")
public TrackDTO handleSelectTrackById(@PathParam("id") Integer id)
{
 final TrackDTO track = TRACKS.get(id);
 if (track == null)
  throw new NotFoundException("Track id [" + id + "] does not found.");
 return track;
}

/**
 * Deletes a track by id.
 */
@DELETE
@Path("/tracks/{id}")
```

```
@Documentation("Deletes a track by id")
public void handleDeleteTrackById(@PathParam("id") Integer id)
{
 if (!TRACKS.containsKey(id))
  throw new NotFoundException("Track id [" + id + "] does not found.");
 TRACKS.remove(id);
}

/**
 * Inserts or updates one or several tracks.
 * <p>
 * The complex response structure corresponds to one of:
 * <ul>
 *  <li>An empty content with the <code>location<code> HTTP header defined
 *   to the access URI.</li>
 *  <li>A JSON array of {@link ResultDetailsDTO} objects.</li>
 * </ul>
 */
@POST
@Path("/tracks")
@Documentation("Inserts or updates one or several tracks")
public Response handleInsertOrUpdateTracks(List<TrackDTO> tracks)
{
 int inserted = 0;
 int updated = 0;

 final ResultDetailsDTO[] resultDetails = new ResultDetailsDTO[tracks.size()];
 int resultIndex = 0;

 final URI base = this.sessionContext.getURIInfoUtility()
   .createBuilderForRESTApplication()
   .path(this.getClass())
   .segment("tracks")
   .build();

 for (final TrackDTO track : tracks)
 {
  final String id = String.valueOf(track.id);
  final URI uri = UriBuilder.fromUri(base).segment(id).build();

  final int code;
  if (TRACKS.containsKey(track.id))
  {
   code = HttpServletResponse.SC_NO_CONTENT;
   updated++;
  }
  else
  {
   code = HttpServletResponse.SC_CREATED;
   inserted++;
  }

  TRACKS.put(track.id, track);

  resultDetails[resultIndex++] = ResultDetailsDTO.create(
   code,
   null,
   String.valueOf(track.id),
   uri);
 }

 if (inserted == 1 && updated == 0)
  return Response.created(resultDetails[0].details).build();

 return Response.ok().entity(resultDetails).build();
}

/**
 * Updates one track.
 */
@PUT
@Path("/tracks/{id}")
@Documentation("Update one track")
public void handleUpdateOneTrack(@PathParam("id") Integer id, TrackDTO aTrack)
{
 final TrackDTO track = TRACKS.get(id);
 if (track == null)
  throw new NotFoundException("Track id [" + id + "] does not found.");

 if (aTrack.id != null && !aTrack.id.equals(track.id))
  throw new BadRequestException("Selected track id [" + id
   + "] is not equals to body track id.");

 TRACKS.put(aTrack.id, aTrack);
}
```

```
}
```

This REST service uses the following Java classes, which represent a Data Transfer Objects (DTO), to serialize and deserialize data:

```
/**
 * DTO for a track.
 */
public final class TrackDTO
{
 public Integer id;
 public String singer;
 public String title;
}

import java.net.*;

/**
 * DTO for result details.
 */
@JsonbPropertyOrder({ "code", "label", "foreignKey", "details" })
public final class ResultDetailsDTO
{
 public int code;
 public String label;
 public String foreignKey;
 public URI details;

 public static ResultDetailsDTO create(
  final int aCode,
  final String aForeignKey,
  final URI aDetails)
 {
  return ResultDetailsDTO.create(aCode, null, aForeignKey, aDetails);
 }

 public static ResultDetailsDTO create(
  final int aCode,
  final String aLabel,
  final String aForeignKey,
  final URI aDetails)
 {
  final ResultDetailsDTO result = new ResultDetailsDTO();
  result.code = aCode;
  result.label = aLabel;
  result.foreignKey = aForeignKey;
  result.details = aDetails;
  return result;
 }
}
```

101.4 Authentication and lookup mechanism

A custom REST service developed with REST Toolkit supports the same authentication methods and lookup mechanism as the built-in REST data services. However, there is a slight difference concerning the 'Anonymous authentication Scheme' since its scope can be wider by using the `AnonymousAccessEnabled`[API]. See REST authentication and permissions [p 750] for more information.

Voir aussi

Authentication [p 676]

Lookup mechanism [p 677]

101.5 REST authentication and permissions

By default, every REST resource Java method requires users to be authenticated.

However, some methods may need to be accessible anonymously. These methods must be annotated by `AnonymousAccessEnabled`[API].

Some methods may need to be restricted to given profiles. These methods may be annotated by `Authorization`[API] to specify an authorization rule. An authorization rule is a Java class that implements the `AuthorizationRule`[API] interface.

```
import javax.ws.rs.*;

import com.orchestranetworks.rest.annotation.*;

/**
 * The REST Toolkit service v1.
 */
@Path("/service/v1")
@Documentation("Service")
public final class Service
{
 ...

 /**
  * Gets service description
  */
 @GET
 @AnonymousAccessEnabled
 public String handleServiceDescription()
 {
  ...
 }

 /**
  * Gets restricted service
  */
 @GET
 @Authorization(IsUserAuthorized.class)
 public RestrictedServiceDTO handleRestrictedService()
 {
  ...
 }
}
```

101.6 **URI builders**

REST Toolkit provides an utility interface `URIInfoUtility`[API] to generate URIs. An instance of this interface is accessible through the injectable built-in object `SessionContext`[API].

101.7 **Exception handling**

A REST Toolkit Java method can produce a standard HTTP error response by throwing a Java exception that extends the JAX-RS class `javax.ws.rs.WebApplicationException`. JAX-RS defines exceptions for various HTTP status codes. EBX defines `UnprocessableEntityException`[API] that adds support for the HTTP `422` *(Unprocessable entity)* code.

Plain Java exceptions are mapped to the HTTP status code `500` *(Internal server error)*.

```
{
  "userMessage": "...", // Mandatory localized message
  "errorCode":   "999", // EBX® error code (optional, used mainly for HTTP error 422)
  "errors": [           // Internal messages useful when debugging (optional).
                        // Usually not displayed to the user and not localized.
     "Message 1", "Message 2" }
  ]
}
```

101.8 **Monitoring**

REST Toolkit events monitoring is similar to the data services log configuration. The difference is the property key which must be `ebx.log4j.category.log.restServices`.

Voir aussi

Monitoring *[p 677]*

Configuring the EBX logs *[p 375]*

Some additional properties are available to configure the log messages. See Configuring REST toolkit services [p 380] for further information.

101.9 Packaging and registration

All applications and components are required to be packaged into the module's Web Application (war file).

The JAX-RS libraries, except the JAX-RS client API, are already included in `ebx.jar` and must not be packaged in the war file.

See Java EE deployment [p 341] for more information.

The registration of a REST Toolkit application is integrated into the EBX module registration process. The registration class must extend `ModuleRegistrationListener`[API], declare the Servlet 3.0 annotation `@WebListener` and override the `handleContextInitialized` method.

See Module registration [p 484] for more information.

```
import javax.servlet.annotation.*;

import com.orchestranetworks.module.*;

@WebListener
public final class RegistrationModule extends ModuleRegistrationListener
{
 @Override
 public void handleContextInitialized(final ModuleInitializedContext aContext)
 {
  // Registers dynamically a REST Toolkit application.
  aContext.registerRESTApplication(RESTApplication.class);
 }
}
```