

ST.96 – ANNEXE VI

RÈGLES ET PRINCIPES DE CONVERSION

Version 6.0

*Révision approuvée par l'Equipe d'experts de la norme XML4IP
du Comité des normes de l'OMPI (CWS) le 3 octobre 2022*

Table des matières

1.	INTRODUCTION	2
1.1	Aperçu	2
1.2	Portée	2
1.3	Comment utiliser le présent document.....	2
1.4	Terminologie	2
1.5	Identificateurs de règle	3
2.	CONSEILS POUR PRÉPARER LES DONNÉES	3
3.	CONSEILS POUR ÉTABLIR LES TABLEAUX DE CORRESPONDANCE	3
4.	CONSEILS POUR CONVERTIR DES DONNÉES.....	4
4.1	Formats de date et d'heure.....	4
4.2	Valeurs booléennes	5
4.3	Codes de langue et de pays	5
4.4	Valeurs des codes et des listes d'énumération	6
4.5	Contraintes d'identité	7
4.6	Différentes structures de données	8
4.7	Erreurs de correspondance entre types.....	9
5.	RÉFÉRENCES	10
	Normes de l'OMPI	10
	Norme sectorielle.....	10
	APPENDICES	11
	APPENDICE A : MISE EN CORRESPONDANCE D'ÉLÉMENTS ET D'ATTRIBUTS.....	11
	APPENDICE B : MISE EN CORRESPONDANCE DE LISTES	11
	APPENDICE C : EXEMPLE DE CODE XSLT.....	12

1. INTRODUCTION

1.1 Aperçu

1. Avant l'adoption de la norme ST.96 de l'OMPI, les offices de propriété intellectuelle utilisaient déjà les normes ST.36, ST.66 et ST.86 de l'OMPI. Les responsables de la norme ST.96 se sont donc efforcés de préserver la possibilité de reprendre des instances de document XML conformes aux normes ST.36, ST.66 et ST.86 (ci-après appelées "normes XML existantes") en les convertissant.

2. Pour faciliter l'échange de données et l'interopérabilité entre un office de propriété industrielle appliquant les normes ST.36, ST.66 et ST.86 et un office appliquant la norme ST.96, il était souhaitable de permettre la conversion dans les deux sens; toutefois, on ne pouvait espérer que cette conversion bidirectionnelle soit parfaite. En effet, compte tenu des améliorations découlant de l'expérience acquise et des progrès technologiques, les structures de la norme ST.96 différaient à de nombreux égards de celles des normes ST.36, ST.66 et ST.86. Le présent document a donc pour objectif de définir dans quelle mesure la conversion bidirectionnelle est possible entre la norme ST.96 et les normes ST.36, ST.66 et ST.86. Il convient d'ailleurs de noter que cette possibilité doit être déterminée au cas par cas.

1.2 Portée

3. Le présent document a pour but d'établir des règles et principes de conversion entre des instances XML de la norme ST.96 et des instances des normes ST.36, ST.66 ou ST.86 de l'OMPI. Il n'a pas pour objet de traiter des conversions à des fins d'application nationale ou des conversions d'instances XML entre différentes versions de la norme ST.96.

4. Ce document contient des tableaux de correspondance entre les éléments et attributs définis dans les différentes normes; ils apparaissent dans l'appendice A. Il contient aussi des tableaux de correspondance entre les valeurs et les codes définis dans les différentes normes; ils apparaissent dans l'appendice B. Tous ces tableaux seront actualisés au fil de l'évolution des normes.

5. Le présent document contient en outre, dans l'appendice C, quelques exemples de conversions effectuées au moyen du langage XSLT (eXtensible Stylesheet Language Transformations), compte tenu des tableaux de correspondance.

1.3 Comment utiliser le présent document

6. Le présent document a pour but d'aider les offices de propriété industrielle à convertir leurs données existant sous un format conforme aux normes ST.36, ST.66 ou ST.86 pour les rendre conformes à la norme ST.96, et vice-versa.

1.4 Terminologie

7. Dans le présent document :

- l'expression "*conversion de données*" s'entend du processus par lequel des données existant sous un format source sont converties en un format cible. Ce processus s'effectue en deux étapes : d'abord une mise en correspondance des données, puis la production du code requis;
- l'expression "*mise en correspondance des données*" s'entend de la mise en correspondance d'éléments ou attributs et de codes ou listes d'énumération entre une source et une cible, ainsi que de toute conversion nécessaire à ce processus. La mise en correspondance d'éléments ou attributs est décrite dans l'appendice A du présent document, tandis que la mise en correspondance de codes ou listes d'énumération est décrite dans l'appendice B;
- l'expression "*production de code*" s'entend de la programmation de ces conversions en langage XSLT, compte tenu des tableaux de correspondance. L'appendice C du présent document contient un exemple de code XSLT;
- l'expression "*instance XML d'entrée*" s'entend de l'instance XML qui doit être convertie; et
- l'expression "*instance XML de sortie*" s'entend de l'instance XML résultant de la conversion.

1.5 Identificateurs de règle

8. Toutes les règles de conversion sont de nature informative et sont désignées par un préfixe de type [TR-*nn*]. La valeur "*nn*" indique le numéro d'ordre séquentiel de la règle. Ainsi, l'identifiant de règle [TR-06] désigne la sixième règle de conversion.

2. CONSEILS POUR PRÉPARER LES DONNÉES

9. En principe, les instances XML créées par un office de propriété industrielle devraient être conformes à la DTD ou au schéma XML qui décrivent les applications de l'office. Cette DTD ou ce schéma peuvent définir certains éléments, attributs, types et espaces de nommage propres à l'office. Pour qu'un office puisse utiliser le module de conversion afin de transformer des données conformes aux normes ST.36, ST.66 ou ST.86 en données conformes à la norme ST.96 et vice-versa, il va probablement devoir modifier ses instances XML de manière à les mettre en conformité avec la norme de l'OMPI concernée.

10. Pour préparer des instances conformes,

- si l'office emploie des noms de composantes qui lui sont propres au lieu des noms définis dans la norme (par exemple `WOApplicationBody`), les noms figurant dans les instances devraient être modifiés pour correspondre aux noms de composantes correspondants de la norme; et
- si aucun espace de nommage n'est déclaré dans les instances, l'office devrait ajouter une déclaration d'espace de nommage conformément aux spécifications de la norme correspondante.

[TR-01] Il convient de vérifier qu'une instance XML d'entrée est conforme à la DTD (fichier `xx-patent-document.dtd` prévu dans la norme ST.36) ou au schéma (normes ST.66, ST.86 et ST.96) correspondants.

3. CONSEILS POUR ÉTABLIR LES TABLEAUX DE CORRESPONDANCE

11. Les tableaux de correspondance sont des éléments essentiels dans le présent document. Ils ont pour but d'établir la relation entre chaque élément et attribut de la norme ST.96 et chaque élément et attribut des normes ST.36, ST.66 et ST.86. Comme nous l'avons expliqué au point 1.1 ci-dessus, cette mise en correspondance n'est pas toujours possible. On trouvera ces tableaux de correspondance dans l'appendice A du présent document.

12. Pour chaque sens de mise en correspondance (c'est-à-dire ST.36 vers ST.96, ST.66 vers ST.96, ST.86 vers ST.96, ST.96 vers ST.36, ST.96 vers ST.66 et ST.96 vers ST.86), il faut définir un fichier différent. Chaque fichier contient deux parties, comme les appendices A et B du présent document).

13. Chaque tableau de correspondance contient les colonnes suivantes :

- Nœud d'entrée;
- Nœud de sortie;
- Type [cardinalité] des nœuds d'entrée et de sortie;
- Condition.

14. Les colonnes du *nœud d'entrée* et du *nœud de sortie* définissent chacun des éléments ou attributs. Dans les composantes ST.96, les noms de balise comportent le préfixe de l'espace de nommage. C'est pourquoi il faut indiquer le chemin complet des éléments ou attributs. Le chemin est représenté par des niveaux hiérarchiques au moyen de la syntaxe XPATH. Les éléments sont énumérés dans l'ordre de leur déclaration dans la DTD ou le schéma correspondant. Dans la syntaxe XPATH, on emploie la barre oblique "/" pour séparer les niveaux hiérarchiques.

15. La colonne *Type [Cardinalité]* permet de désigner :

- le type employé pour les éléments ou attributs. Étant donné que la norme ST.36 est définie par des DTD, seuls les types suivants peuvent être employés : ID, CDATA et #PCDATA. S'il s'agit des normes ST.66, ST.86 et ST.96, la colonne peut indiquer des types prédéfinis par le W3C, mais aussi des types définis par l'utilisateur. S'il s'agit de composantes ST.96 ou d'autres normes externes invoquées, les types de données doivent comporter le préfixe de l'espace de nommage; et

- la cardinalité des éléments ou attributs :
 - 1..1 = une seule occurrence obligatoire;
 - 0..1 = une seule occurrence facultative;
 - 1..n(∞) = une ou plusieurs occurrences obligatoires;
 - 0..n(∞) = une ou plusieurs occurrences facultatives.

16. La colonne *Condition* contient des instructions ou des règles particulières de mise en correspondance ou de conversion concernant la source ou la cible, qui sont décrites plus loin dans la partie intitulée "Conseils pour convertir des données".

Nœud d'entrée	Type [Cardinalité]	Condition	Nœud de sortie	Type [Cardinalité]
ReportCitation/CitedReference/@id	xsd:token [0..1]	TR-23	citation/@id	ID [0..1]

Exemple de tableau de correspondance

- [TR-02] Un tableau de correspondance DEVRAIT être établi pour chaque schéma concernant les niveaux de document défini dans la norme ST.96, par exemple l'élément `ApplicationBody_V6_0.xsd`, afin de faciliter la conversion de données.
- [TR-03] Les offices DEVRAIENT faire en sorte que les conversions bidirectionnelles garantissent la préservation de l'intégrité des données, et que les limites soient entièrement décrites et prises en charge (par exemple si certaines données sont perdues au cours de la conversion).

4. CONSEILS POUR CONVERTIR DES DONNÉES

17. Outre les tableaux établissant la correspondance élément par élément, il convient aussi d'examiner la conversion de données. En effet, les formats de données employés dans les éléments ou attributs et dans la structure de leurs contenus conformément à la norme ST.96 peuvent différer des éléments ou attributs correspondants prévus dans les normes ST.36, ST.66 ou ST.86. De nombreux éléments ou attributs de la norme ST.96 proviennent des normes ST.36, ST.66 ou ST.86, mais ont été redéfinis : la structure des données a été simplifiée, ou de nouvelles technologies XML ont été employées, par exemple en appliquant une méthode de conception orientée vers les données. Des conseils sont donc proposés ci-après pour prendre en compte un certain nombre de points avant de convertir des instances XML; ces questions sont abordées plus en détail dans les parties suivantes :

- les formats de date et d'heure;
- les valeurs booléennes;
- les codes de langue et de pays;
- les valeurs des codes et des listes d'énumération;
- les contraintes d'identité;
- les différentes structures de données; et
- les erreurs de correspondance entre types.

4.1 Formats de date et d'heure

18. Les DTD de la norme ST.36 ne définissent pas de format strict de date en raison de leurs limites intrinsèques. Néanmoins, il est recommandé d'employer le format "AAAAMMJJ". Dans la norme ST.36, la date relève du type `#PCDATA`, bien que le format jour/mois/année soit recommandé pour certaines dates, par exemple une date de postage ou de priorité. Dans les normes ST.66, ST.86 et ST.96, les dates sont définies conformément à l'élément `xsd:date`, qui exige le format AAAA-MM-JJ.

19. Tous les champs de date pour lesquels il est obligatoire d'indiquer le jour, le mois et l'année peuvent accepter un format conforme à la norme ST.96. Dans ce cas, il convient d'appliquer les règles de conversion suivantes :

[TR-04] Pour convertir la date au format ST.96, il faut convertir la valeur du champ de date ST.36, puis la copier dans le champ de date ST.96.

Exemple : Entrée : valeur du champ de date ST.36 = 20081025; format du champ de date : YYYYMMDD ; Sortie : valeur du champ de date ST.96 = 2008-10-25.

[TR-05] Pour convertir la date au format ST.36, il faut copier la valeur du champ de date ST.96 dans le champ de date ST.36 après en avoir retiré les tirets.

Exemple : Entrée : valeur du champ de date ST.96 = 2008-10-25; Sortie : valeur du champ de date ST.36 = 20081025.

20. La norme ST.36 ne définit pas de format strict d'heure en raison de ses limites intrinsèques. Selon cette norme, l'heure relève du type #PCDATA, bien que le format HHMM soit recommandé. Dans les normes ST.66, ST.86 et ST.96, les champs d'heure sont définis conformément à l'élément `xsd:time`, qui exige le format hh:mm:ss.

21. Tous les champs d'heure pour lesquels il est obligatoire d'indiquer l'heure, la minute et la seconde peuvent prendre un format conforme à la norme ST.96. Dans ce cas, il convient d'appliquer les règles de conversion suivantes :

[TR-06] Pour convertir l'heure au format ST.96, il faut convertir la valeur du champ d'heure ST.36, puis la copier dans le champ d'heure ST.96.

Exemple : Entrée : valeur du champ d'heure ST.36 = 1030 (format HHMM);
Sortie : valeur du champ d'heure ST.96 = 10:30:00.

[TR-07] Pour convertir l'heure au format ST.36, il faut copier la valeur du champ d'heure ST.96 dans le champ d'heure ST.36 après en avoir retiré les deux points et les secondes.

Exemple : Entrée : valeur du champ d'heure ST.96 = 10:30:00; Sortie : valeur du champ d'heure ST.36 = 1030.

4.2 Valeurs booléennes

22. La norme ST.36 ne définit pas de format strict pour les valeurs booléennes en raison des limites intrinsèques des DTD. En principe, chaque office a établi ses propres règles pour faire en sorte que le format des valeurs booléennes soit défini de manière cohérente dans ses instances XML. Les valeurs suivantes sont couramment employées dans les variables booléennes des instances XML conformes à la norme ST.36 : "0 / 1", "no / yes" et "false / true". Dans la norme ST.96, ce sont les valeurs "false / true" qui sont utilisées. Pour convertir des valeurs booléennes d'instances ST.36 en valeurs booléennes d'instances ST.96, il convient d'appliquer les règles suivantes :

[TR-08] Pour une conversion vers une instance ST.36, il faut copier la valeur booléenne du champ ST.96 dans le champ ST.36. Au besoin, l'office peut adapter les valeurs en fonction de ses propres pratiques.

[TR-09] Pour une conversion vers une instance ST.96, il faut convertir la valeur booléenne du champ ST.36 en valeur "true" ou "false" puis la copier dans le champ ST.96.

23. Pour faciliter la conversion des valeurs booléennes, un système de conversion est proposé dans l'exemple de code XSLT figurant dans l'appendice C : il permet de transformer les valeurs "0 / 1", "no / yes" et "false / true" d'une instance XML ST.36 en valeurs "false / true" dans l'instance ST.96. Si l'instance ST.36 contient d'autres valeurs, l'office devrait convertir ces valeurs en l'une des valeurs communes avant d'exécuter le script XSLT.

4.3 Codes de langue et de pays

24. Dans la norme ST.36, l'élément ou l'attribut de langue est une chaîne alphanumérique (#PCDATA); toutefois, il est recommandé d'employer le code ISO à deux lettres pour décrire cet élément ou attribut. Dès lors, si ce code à deux lettres n'a pas été employé dans les instances XML ST.36, le code de langue devra être converti en code à deux lettres avant d'effectuer la conversion en instance ST.96.

25. Dans la norme ST.36, l'élément ou l'attribut de pays est une chaîne alphanumérique (#PCDATA); toutefois, il est recommandé d'employer le code ISO à deux lettres pour décrire cet élément ou attribut. Si les deux normes emploient ce code à deux lettres, la norme ST.36 autorise les minuscules tandis que la norme ST.96 impose les majuscules. Il convient donc de modifier la casse du code en conséquence avant d'effectuer la conversion.

- [TR-10] Pour une conversion vers une instance ST.96, le code de langue ST.36 doit être mis en majuscules avant d'être copié dans le champ de langue ST.96.
- [TR-11] Pour une conversion vers une instance ST.36, le code de langue ST.96 doit être copié dans le champ de langue ST.36.

4.4 Valeurs des codes et des listes d'énumération

26. Les normes ST.36, ST.66, ST.86 et ST.96 contiennent des jeux de codes et différentes listes d'énumération. Les codes ou listes ST.96 sont souvent identiques à ceux des trois autres normes. Cependant, il arrive que la norme ST.96 définisse des codes ou listes supplémentaires ou différents pour respecter ses règles et conventions de conception, ou pour permettre de prendre en compte les pratiques propres à chaque office. Le présent document contient quelques conseils sur la conversion de ces codes, selon que la norme ST.96 prenne en charge les codes des trois autres normes ou qu'elle prévoise des codes supplémentaires.

27. Certains champs ST.36 sont de type #PCDATA et non des listes de valeurs spécifiques, tandis que la norme ST.96 prévoit généralement des codes ou des listes d'énumération. Dans les normes ST.66 et ST.86, certains types sont constitués par l'union d'un format libre et d'une liste.

- [TR-12] Pour une conversion vers une instance ST.96, il faut établir la correspondance entre la valeur des champs ST.66 ou ST.86 et les codes ou listes ST.96, puis le code correspondant doit être copié dans le champ ST.96. Il faut ignorer les valeurs de format libre, sauf si une correspondance avec un code ST.96 peut être trouvée.
- [TR-13] Pour une conversion vers une instance ST.96, il faut ignorer les valeurs des champs ST.36, sauf si une correspondance avec un code ST.96 peut être trouvée. Si une valeur n'est pas conforme au type de données prévu dans la norme ST.96, il sera peut-être nécessaire de la reformater afin que l'instance puisse être validée.
- [TR-14] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, il faut établir la correspondance entre le code ST.96 et le champ de la norme concernée.

ATTENTION : Lorsqu'on copie des valeurs provenant de champs ST.36, ST.66 ou ST.86 vers un champ ST.96 (ou vice-versa), ou lorsqu'on modifie des instances ST.36, ST.66 ou ST.86 pour leur intégrer des valeurs provenant de champs ST.96, il faut vérifier que les nouvelles valeurs ajoutées ne contreviennent pas à une règle de fonctionnement du nouveau contexte.

Jeu de valeurs identiques

28. Dans la plupart des cas, les normes ST.36, ST.66, ST.86 et ST.96 prévoient le même jeu de codes ou de listes d'énumération pour les éléments ou attributs énumérés dans l'appendice B du présent document.

Par exemple :

Composante ST.66	Valeurs autorisées dans la norme ST.66	Composante ST.96	Valeurs autorisées dans la norme ST.96
MarkKind	Individual	MarkKind	Individual
	Collective		Collective
	Certificate		Certificate
	Guarantee		Guarantee
	Defensive		Defensive
	Other		Other

29. Dans ce cas, les recommandations suivantes sont d'application :

- [TR-15] Pour une conversion vers une instance ST.96, il faut copier la valeur du champ ST.36, ST.66 ou ST.86 dans le champ ST.96 correspondant. Le format de la valeur doit être conforme aux règles et conventions de conception de la norme ST.96.

[TR-16] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, il faut copier la valeur du champ ST.96 dans le champ ST.36, ST.66 ou ST.86. Le format de la valeur doit être conforme aux conventions énoncées dans la norme cible correspondante.

30. Si des éléments ou attributs sont obligatoires dans la norme ST.96, mais facultatifs dans les normes ST.36, ST.66 ou ST.86, il convient d'appliquer les règles suivantes :

[TR-17] Si la composante ST.36, ST.66 ou ST.86 contient des valeurs, il faut copier la valeur de chaque champ ST.36, ST.66 ou ST.86, dans le champ ST.96 correspondant. Le format de la valeur doit être conforme aux règles et conventions de conception de la norme ST.96.

[TR-18] Si la composante ST.36, ST.66 ou ST.86 ne contient aucune valeur, il faut remplir les champs ST.96 correspondants avec la valeur "Undefined".

31. Aucun élément ou attribut obligatoire dans les normes ST.36, ST.66 ou ST.86 n'est facultatif dans la norme ST.96. Le présent document ne contient donc aucune règle concernant ce cas de figure.

Jeu de valeurs différents

32. Dans certains cas énumérés à l'appendice B du présent document, la norme ST.96 prévoit des codes ou des listes d'énumération différents des normes ST.36, ST.66 ou ST.86, alors que ces codes ou listes ont la même signification. Par exemple :

Composante ST.36	Valeurs autorisées dans la norme ST.36	Composante ST.96	Valeurs autorisées dans la norme ST.96
Orient	Port	orientation	Portrait
	Land		Landscape

33. Dans certains cas, la norme ST.96 prévoit plus de codes ou de listes que les normes ST.36, ST.66 ou ST.86. La conversion depuis ces trois normes vers la norme ST.96 ne pose alors aucune difficulté. En revanche, la conversion depuis la norme ST.96 vers les trois autres n'est pas garantie. Pour que cette conversion se déroule correctement, il faut modifier les normes XML existantes afin de prendre en compte les codes ou listes supplémentaires de la norme ST.96. Par exemple :

Composante ST.36	Valeurs autorisées dans la norme ST.36	Composante ST.96	Valeurs autorisées dans la norme ST.96
Color	color	ColourMode	Colour
	Bw		Black and white
	-		Greyscale

34. En général, une composante ST.96 ne définit pas moins de valeurs que la composante correspondante de la norme ST.36, ST.66 ou ST.86. Le présent document ne contient donc aucune règle concernant ce cas de figure.

[TR-19] Pour une conversion vers une instance ST.96, la valeur doit être copiée depuis le champ ST.36, ST.66 ou ST.86 dans le champ ST.96, dès lors que les deux valeurs ont la même signification.

[TR-20] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, la valeur doit être copiée depuis le champ ST.96 dans le champ ST.36, ST.66 ou ST.86, dès lors que les deux valeurs ont la même signification.

[TR-21] S'il n'existe pas de valeur correspondante dans la norme cible, la valeur doit être copiée telle quelle dans l'instance de sortie, même si elle n'est pas conforme au schéma ou la DTD de cette norme.

4.5 Contraintes d'identité

35. Il est recommandé dans la norme ST.96 d'employer les types `xsd:key`, `xsd:unique` ou `xsd:keyref`, ainsi que `xsd:ID`, `xsd:IDREF` ou `xsd:IDREFS` pour définir les contraintes d'identité, alors que la norme ST.36 prévoit l'emploi des éléments `ID`, `IDREF` ou `IDREFS`. Certains attributs de la norme ST.36 emploient aussi les types `ID` et `IDREF` ou `IDREFS` : ainsi, l'élément `citation/id` emploie le type `ID`. Les valeurs commençant par un caractère numérique ne sont pas autorisées dans les champs `xsd:ID`, `xsd:IDREF` ou `xsd:IDREFS`.

[TR-22] Pour une conversion vers une instance ST.96, la valeur des champs ST.36, ST.66 ou ST.86 associés aux types `xsd:ID`, `xsd:IDREF` ou `xsd:IDREFS` doit être copiée dans les champs ST.96 correspondants.

[TR-23] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, la valeur des champs ST.96 associés aux types `xsd:ID`, `xsd:IDREF` ou `xsd:IDREFS` doit être copiée dans les champs ST.36, ST.66 ou ST.86 correspondants. Il faut aussi copier les valeurs de type `xsd:key`, `xsd:unique` ou `xsd:keyref` des champs ST.96 dans les champs ST.36, ST.66 ou ST.86 en s'assurant de ne pas dupliquer les valeurs `ID` dans l'instance et d'adapter les valeurs `IDREF` et `IDREFS` correspondantes. Les valeurs de type `xsd:key`, `xsd:unique` ou `xsd:keyref` commençant par un caractère numérique doivent être converties en caractères non numériques.

4.6 Différentes structures de données

36. La norme ST.96 prévoit différentes structures pour intégrer certaines composantes des normes ST.36, ST.66 ou ST.86. Quelques conseils de conversion sont proposés dans le présent document pour passer d'une structure à l'autre. Trois interventions peuvent être effectuées dans ce contexte : supprimer, ajouter et renommer.

Supprimer

37. Certains éléments ou attributs définis dans les normes ST.36, ST.66 ou ST.86 ont été supprimés dans la norme ST.96, car ils ne sont plus utilisés. Ils sont signalés par la mention "NOT USED" (non utilisé) dans l'appendice A du présent document. C'est par exemple le cas des éléments "citation", "nplcit", "article", "book" et "text", qui n'ont donc aucun élément correspondant dans la norme ST.96.

[TR-24] Pour une conversion vers une instance ST.96, il faut ignorer les champs ST.36, ST.66 ou ST.86 puisque ces éléments n'ont pas de contrepartie dans la norme ST.96.

[TR-25] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, il est inutile de reprendre les valeurs des champs ST.36, ST.66 ou ST.86, car ces champs sont facultatifs dans les trois normes.

Ajouter

38. Certains éléments ou attributs sont nouveaux dans la norme ST.96 et n'ont donc pas de contrepartie dans les normes ST.36, ST.66 ou ST.86.

[TR-26] Tous les éléments propres à l'office de la propriété industrielle doivent être ignorés dans le processus de conversion.

[TR-27] Si un élément ou attribut supplémentaire a été ajouté, pour une conversion vers une instance ST.96, il ne faut pas renseigner le champ ST.96, car il n'existe pas de contrepartie dans les champs ST.36, ST.66 ou ST.86.

[TR-28] Lorsqu'on ajoute un élément ou attribut supplémentaire, il ne faut pas créer d'élément obligatoire si l'élément facultatif correspondant n'est pas présent dans l'instance d'entrée. Un message d'erreur devrait être affiché au cours du processus de validation de l'instance de sortie. L'office peut alors décider de réconcilier les éléments au cas par cas.

Renommer

39. Presque tous les champs ont un nouveau nom dans la norme ST.96. On trouvera dans l'appendice A du présent document un tableau de correspondance établi champ par champ. Ainsi, l'élément `absno` de la norme ST.36 correspond à l'élément `AbstractNumber` de la norme ST.96.

[TR-29] Pour une conversion vers une instance ST.96, la valeur des champs ST.36, ST.66 ou ST.86 doit être copiée dans les champs ST.96.

- [TR-30] Pour une conversion vers une instance ST.36, ST.66 ou ST.86, la valeur des champs ST.96 doit être copiée dans les champs ST.36, ST.66 ou ST.86.

Changer l'ordre des éléments dans une création de séquence

40. Dans un schéma ST.96, certains éléments enfants ont un ordre différent de celui qui est défini dans un schéma ST.36, ST.66 ou ST.86. Or l'ordre des éléments enfants est important dans une création de séquence; le changement de l'ordre des éléments enfants dans une séquence doit donc être pris en compte au moment de la conversion.

Élément vide

41. Comme les éléments vides ne sont pas autorisés dans la norme ST.96, il est impossible d'établir une correspondance un à un avec des éléments vides de la norme ST.36, ST.66 ou ST.86. Ces derniers ayant été définis de différentes manières dans les trois normes, leur conversion sera différente et doit être traitée au cas par cas.

42. Il arrive que certains éléments ST.36 destinés à signaler l'existence d'informations soient en réalité vides. Dans ce cas, les éléments ST.96 correspondants sont définis selon le type `xsd:boolean`. Dans d'autres cas, la présence de certains éléments vides dans une instance XML ST.36 peut être mise en correspondance avec une valeur d'énumération dans l'instance XML ST.96.

43. Dans le schéma de modèle ST.66 comme dans celui de la norme ST.86, la plupart des éléments ont obligatoirement un élément enfant, ce qui signifie que les éléments parents peuvent être vides. Pour éviter ces contenus vides, il faut remplacer la structure d'élément `sequence` par une structure multiple `choice` dans les éléments ST.96 correspondants. Ce type de changement structurel doit être prévu dès la conception de la conversion.

Changement de cardinalité

44. Un élément considéré comme facultatif dans une norme peut devenir obligatoire dans une autre.

- [TR-31] Si la cardinalité d'un élément est modifiée de telle sorte que celui-ci passe de facultatif à obligatoire, il ne faut pas créer d'élément obligatoire si l'élément facultatif correspondant n'est pas présent dans l'instance d'entrée. Un message d'erreur devrait être affiché au cours du processus de validation de l'instance de sortie. L'office peut alors décider de réconcilier les éléments au cas par cas.

Changement de structure

45. La structure définie dans la norme ST.96 est parfois différente de celle qui est définie dans les normes ST.36, ST.66 ou ST.86 afin d'être plus claire. Lorsque ce cas se produit, il convient d'établir quelques règles de conversion au cas par cas. Par exemple,

- si l'élément `doc-page` est employé dans un contexte d'images, il faut vérifier l'attribut du type :
 - si `@type= jpg` ou `tif`, il faut établir la correspondance avec l'élément `PageImage`;
 - si `@type= jpg`, il faut établir la correspondance avec l'élément `DocumentURI`;
- dans les autres contextes, il faut vérifier la présence de l'attribut `ocr` :
 - si un attribut `ocr` est défini, il faut établir la correspondance avec l'élément `DocumentURI`;
 - dans les autres cas, il faut établir la correspondance avec l'élément `PageImage`.

4.7 Erreurs de correspondance entre types

46. Dans la norme ST.96, certains éléments ou attributs sont soumis à des restrictions de type, tandis que la DTD de la norme ST.36 dispose de très peu d'attributs ou d'éléments typés. Seuls les types `#PCDATA`, `ID`, `IDREF` et `CDATA` sont employés pour des éléments ou des attributs individuels.

Contraintes de type "Pattern"

47. Certains champs ST.36 sont définis selon le type #PCDATA et ne comportent donc pas de restriction définie au moyen de la facette "Pattern". Ces champs peuvent accepter sans problème les valeurs plus restreintes définies dans la norme ST.96.

- [TR-32] Pour une conversion vers une instance ST.96, la valeur du champ ST.36 doit être copiée telle quelle dans le champ ST.96. Si cette valeur n'est pas conforme aux contraintes de type "Pattern" définies dans la norme ST.96, elle peut être reformatée afin que l'instance puisse être validée.
- [TR-33] Pour une conversion vers une instance ST.36, la valeur du champ ST.96 doit être copiée telle quelle dans le champ de texte libre prévu par la norme ST.36.

Types de données prédéfinis par le W3C

48. En principe, les éléments ou attributs individuels dépourvus de type qui sont définis dans une DTD sont mis en correspondance avec l'élément `xsd:string`, pour lequel il est obligatoire de définir un type. La norme ST.96 utilise les types de données suivants, qui ont été prédéfinis par le W3C : `xsd:token`, `xsd:positiveInteger`, `xsd:boolean` et `xsd:string`. Bien que le format défini pour certains champs ST.36 soit alphanumérique (#PCDATA), les valeurs présentes dans ces champs devraient normalement être conformes aux types de données prédéfinis par le W3C et repris dans la norme ST.96.

- [TR-34] Pour une conversion vers une instance ST.96, la valeur du champ ST.36 doit être copiée telle quelle dans le champ ST.96. Si cette valeur n'est pas conforme au type de données prévu dans la norme ST.96, elle peut être reformatée afin que l'instance puisse être validée.
- [TR-35] Pour une conversion vers une instance ST.36, la valeur du champ ST.96 doit être copiée telle quelle dans le champ de texte libre prévu par la norme ST.36.

5. RÉFÉRENCES

Normes de l'OMPI

- Norme [ST.36](#) de l'OMPI Utilisation du XML dans le traitement de l'information en matière de brevets ;
- Norme [ST.66](#) de l'OMPI Utilisation du XML dans le traitement de l'information en matière de marques ;
- Norme [ST.86](#) de l'OMPI Utilisation du langage XML dans le traitement de l'information en matière de dessins et modèles industriels.

Norme sectorielle

- W3C XSLT : <https://www.w3.org/TR/xslt/>

APPENDICES

Les présents appendices sont fondés sur la version 6.0 de l'annexe III de la norme ST.96 de l'OMPI (XML Schema V6_0). Ils peuvent être consultés dans leur intégralité ici ([AnnexVI Appendices A B C V6 0.zip](#)).

APPENDICE A : MISE EN CORRESPONDANCE D'ÉLÉMENTS ET D'ATTRIBUTS

On trouvera dans le présent appendice A un modèle permettant de mettre en correspondance un à un les éléments et attributs de la norme ST.96 avec les éléments et attributs correspondants des normes ST.36, ST.66 ou ST.86. Il n'est pas toujours possible d'effectuer cette mise en correspondance pour les raisons exposées dans l'Annexe VI de la norme ST.96 intitulée "Règles et principes de conversion". Le but du présent appendice A est donc de proposer une mise en correspondance, dans la mesure du nécessaire, entre la norme ST.96 et les normes ST.36, ST.66 ou ST.86.

Les tableaux de correspondance suivants se trouvent dans le fichier correspondant :

- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `application-body` de la version ST.36 vers l'élément `ApplicationBody` de la version ST.96;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `ApplicationBody` de la version ST.96 vers l'élément `application-body` de la version ST.36;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `bibliographic-data` de la version ST.36 vers l'élément `BibliographicData` de la version ST.96;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `BibliographicData` de la version ST.96 vers l'élément `bibliographic-data` de la version ST.36;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `Transaction` de la version ST.66 vers l'élément `TrademarkTransaction` de la version ST.96;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `TrademarkTransaction` de la version ST.96 vers l'élément `Transaction` de la version ST.66;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `Transaction` de la version ST.86 vers l'élément `DesignTransaction` de la version ST.96;
- Tableau de correspondance des éléments et attributs concernant la conversion de l'élément `DesignTransaction` de la version ST.96 vers l'élément `Transaction` de la version ST.86.

APPENDICE B : MISE EN CORRESPONDANCE DE LISTES

On trouvera dans le présent appendice B un modèle permettant de mettre en correspondance un à un les codes ou listes d'énumération de la norme ST.96 avec les codes ou listes d'énumération correspondants des normes ST.36, ST.66 ou ST.86. Il n'est pas toujours possible d'effectuer cette mise en correspondance pour les raisons exposées dans l'Annexe VI de la norme ST.96 intitulée "Règles et principes de conversion".

Les tableaux de correspondance suivants se trouvent dans le fichier correspondant :

- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `application-body` de la version ST.36 vers l'élément `ApplicationBody` de la version ST.96;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `ApplicationBody` de la version ST.96 vers l'élément `application-body` de la version ST.36;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `bibliographic-data` de la version ST.36 vers l'élément `BibliographicData` de la version ST.96;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `BibliographicData` de la version ST.96 vers l'élément `bibliographic-data` de la version ST.36;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `Transaction` de la version ST.66 vers l'élément `TrademarkTransaction` de la version ST.96;

- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `TrademarkTransaction` de la version ST.96 vers l'élément `Transaction` de la version ST.66;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `Transaction` de la version ST.86 vers l'élément `DesignTransaction` de la version ST.96;
- Tableau de correspondance de la liste d'énumération concernant la conversion de l'élément `DesignTransaction` de la version ST.96 vers l'élément `Transaction` de la version ST.86.

APPENDICE C : EXEMPLE DE CODE XSLT

On trouvera dans le présent appendice C un exemple de code écrit dans le langage XSLT (*Extensible Stylesheet Language Transformations*) qui permet de convertir des données entre une instance ST.96 et une instance ST.36, ST.66 ou ST.86, compte tenu des appendices A et B.

- Exemple de code XSLT traitant les éléments "ApplicationBody", "BibliographicData", "TrademarkTransaction" et "DesignTransaction" : les feuilles de style de conversion se composent d'un ensemble de fichiers permettant de convertir des instances ST.96 en instances ST.36, ST.66 ou ST.86 et vice-versa.

[L'annexe VII suit]