

# RexPublisher V1

## Manuel de programmation Des feuilles de styles

**Auteur** : Dominique SCIAMMA  
**Date de modification** : 17/06/2005  
**Version** : 8

© Cervoni Conseil  
Email : info@rexpublisher

## Sommaire

<b>0</b>	<b>Préambule .....</b>	<b>1</b>
<b>1</b>	<b>Qu'est-ce qu'une feuille de style ? .....</b>	<b>2</b>
<b>2</b>	<b>Où sont rangées les feuilles de style ? .....</b>	<b>2</b>
<b>3</b>	<b>Feuille de style et génération .....</b>	<b>2</b>
<b>4</b>	<b>Quels sont les différents type de styles ?.....</b>	<b>2</b>
<b>5</b>	<b>Quels sont les documents publiables ?.....</b>	<b>3</b>
<b>6</b>	<b>Comment nommer les feuilles de style ?.....</b>	<b>3</b>
6.1	Les paramètres de publication .....	3
6.2	Le sommaire général :.....	3
6.3	Les sommaires de Sections .....	4
6.4	Les sommaires de Rubrique .....	4
6.5	Les sommaires de sous-Rubriques.....	4
6.6	Les sommaires de Themes .....	4
6.7	Les sommaires de Mots-clé .....	5
6.8	Les sommaires d'articles.....	5
<b>7</b>	<b>Comment est structurée une feuille de style.....</b>	<b>6</b>
7.1	Qu'est ce qu'une Balise RexPublisher .....	6
7.2	Les Sous-balises RexPublisher.....	6
7.2.1	<i>Spécialisation par Section .....</i>	<i>6</i>
7.2.2	<i>Spécialisation par Section et Rubrique.....</i>	<i>6</i>
7.2.3	<i>Spécialisation par Section, Rubrique, et Sous-Rubrique.....</i>	<i>7</i>
7.2.4	<i>Quelques finesses sur les corps de balises.....</i>	<i>7</i>
7.3	Corps de sous-balises.....	8
7.4	Variables RexPublisher .....	8
7.4.1	<i>Types de variables .....</i>	<i>8</i>
7.4.2	<i>Syntaxe.....</i>	<i>8</i>
7.5	Balises Obligatoires.....	10
7.5.1	<i>Balise &lt;!<b>@DOC</b>&gt;.....</i>	<i>10</i>
7.5.2	<i>Balise &lt;!<b>@NAME</b>&gt;.....</i>	<i>10</i>
7.5.3	<i>Balise &lt;!<b>@EXTENTION</b>&gt; .....</i>	<i>10</i>
7.5.4	<i>Balise &lt;!<b>@PATH</b>&gt; .....</i>	<i>10</i>
7.6	Balises Optionnelles.....	10
7.6.1	<i>Balise &lt;!<b>@IMGPATH</b>&gt;.....</i>	<i>10</i>

7.6.2	Balise <! <i>@FILEPATH</i> >.....	10
7.6.3	Balise <! <i>@INSERTEDIMGPATH</i> >.....	10
7.6.4	Balise <! <i>@BRKSTY</i> >.....	10
7.6.5	Balise <! <i>@HREFPAT(Type)</i> >.....	10
7.6.6	Balise <! <i>@_HIDEHREF_</i> >.....	11
7.7	Balises de fichiers associés .....	11
7.7.1	Balise <! <i>@INCLUDE</i> >.....	11
7.7.2	Balise <! <i>@ATTACHED TO</i> >.....	11
7.8	Balises de Publication .....	11
7.8.1	Balise <! <i>@PUBLICATION</i> >.....	11
0.1.1	Exemples :.....	12
<b>8</b>	<b>Comment est interprétée une feuille de style .....</b>	<b>14</b>
8.1	Recherche de la feuille de style .....	14
8.1.1	Le sommaire général.....	14
8.1.2	Les sommaires de Section .....	14
8.1.3	Les sommaires de Rubriques.....	14
8.1.4	Les sommaires de sous-rubriques.....	14
8.1.5	Les Articles unitaires.....	15
8.1.6	Les sommaires par Thèmes .....	15
8.1.7	Les sommaires par Mot-clé .....	15
8.2	Ecriture du document .....	16
8.2.1	Constitution du fichier .....	16
8.2.2	Recherche de la balise ! <i>@DOC</i> .....	16
8.2.3	Recherche de la valeur d'une variable .....	16
<b>9</b>	<b>Liste des Variables et fonctions RexPublisher.....</b>	<b>18</b>
9.1	Variables en Dur.....	18
9.1.1	<i>PUBLICATION</i> .....	18
9.1.2	<i>EDITION.NB</i> .....	18
9.1.3	<i>EDITION.DATE</i> ou <i>DATE</i> .....	18
9.1.4	<i>RUB</i> .....	18
9.1.5	<i>SRB</i> .....	18
9.1.6	<i>SSRB</i> .....	18
9.1.7	<i>SRT</i> .....	18
9.1.8	<i>THM</i> .....	18
9.1.9	<i>TIT</i> .....	18
9.1.10	<i>CHA</i> .....	18
9.1.11	<i>TXT</i> .....	18

9.1.12	SIG .....	18
9.1.13	KWDLIST.....	18
9.2	Variables en Dur conditionnelle.....	18
9.3	Variables en Dur spéciales.....	18
9.3.1	RUB.IDX.....	19
9.3.2	CURR_RUB.IDX.....	19
9.3.3	RUB.ORDER.....	19
9.3.4	SRB.IDX.....	19
9.3.5	CURR_SRB.IDX.....	19
9.3.6	SRB.ORDER.....	20
9.3.7	SSRB.IDX.....	20
9.3.8	CURR_SSRB.IDX.....	20
9.3.9	SSRB.ORDER.....	20
9.3.10	ART.INTNB.....	20
9.3.11	ART.LOCNB.....	21
9.3.12	ART.POS(Profondeur).....	21
9.4	Variables relatives aux Signatures.....	22
9.4.1	SIG.NBR.....	22
9.4.2	SIG.NAME, SIG(x).NAME.....	22
9.4.3	SIG.FIRSTNAME, SIG(x).FIRSTNAME.....	22
9.4.4	SIG.ALIAS SIG(x).ALIAS.....	22
9.4.5	SIG.CIVILITY, SIG(x).CIVILITY.....	22
9.4.6	SIG.TITLE, SIG(x).TITLE.....	22
9.4.7	SIG.INITIALS, SIG(x).INITIALS.....	22
9.4.8	SIG.EMAIL, SIG(x).EMAIL.....	22
9.5	Variables relatives aux Images.....	23
9.5.1	IMG.NBR.....	23
9.5.2	IMG.NAME, IMG(x).NAME.....	23
9.5.3	IMG.AUTHOR, IMG(x).AUTHOR.....	23
9.5.4	IMG.COPYRIGHT, IMG(x).COPYRIGHT.....	23
9.5.5	IMG.LINK, IMG(x).LINK.....	23
9.5.6	IMG.CAPTION, IMG(x).CAPTION.....	23
9.5.7	IMG.EXTENTION, IMG(x).EXTENTION.....	23
9.5.8	IMG.WIDTH, IMG(x).WIDTH.....	23
9.5.9	IMG.HEIGHT, IMG(x).HEIGHT.....	23
9.5.10	IMG.INDEX.....	23
9.6	Variables relatives aux Fichiers.....	24

9.6.1	<i>FILE.NBR</i> .....	24
9.6.2	<i>FILE.NAME, FILE(x).NAME</i> .....	24
9.6.3	<i>FILE.AUTHOR, FILE(X).AUTHOR</i> .....	24
9.6.4	<i>FILE.COPYRIGHT, FILE(x).COPYRIGHT</i> .....	24
9.6.5	<i>FILE.CAPTION, FILE(X).CAPTION</i> .....	24
9.6.6	<i>FILE.SIZE, FILE(X).SIZE</i> .....	24
9.6.7	<i>FILE.EXTENTION, FILE(X).EXTENTION</i> .....	24
9.6.8	<i>FILE.INDEX, FILE(X).INDEX</i> .....	24
9.6.9	<i>FILE.ORIGINAL, FILE(X).ORIGINAL</i> .....	24
9.7	Variables relatives aux Liens.....	25
9.7.1	<i>LNK.NBR</i> .....	25
9.7.2	<i>LNK.RUB, LNK(x).RUB</i> .....	25
9.7.3	<i>LNK.SRB, LNK(x).SRB</i> .....	25
9.7.4	<i>LNK.SSRB, LNK(x).SSRB</i> .....	25
9.7.5	<i>LNK.ARTINTNB, LNK(x).ART.INTNB</i> .....	25
9.7.6	<i>LNK.SRT, LNK(x).SRT</i> .....	25
9.7.7	<i>LNK.THM, LNK(x).THM</i> .....	25
9.7.8	<i>LNK.TIT, LNK(x).TIT</i> .....	25
9.7.9	<i>LNK.LEADIN, LNK(x).LEADIN</i> .....	25
9.7.10	<i>LNK.TXT, LNK(x).TXT</i> .....	25
9.7.11	<i>LNK.SIG.XXX, LNK(x).SIG.XXX, LNK.SIG(y).XXX, LNK(x).SIG(y).XXX</i> .....	25
9.8	Variables relatives aux encadrés .....	26
9.8.1	<i>FRAME.NBR</i> .....	26
9.8.2	<i>FRAME.RUB, FRAME(x).RUB</i> .....	26
9.8.3	<i>FRAME.SRB, FRAME(x).SRB</i> .....	26
9.8.4	<i>FRAME.SSRB, FRAME(x).SSRB</i> .....	26
9.8.5	<i>FRAME .ART.INTNB, FRAME(x).ART.INTNB</i> .....	26
9.8.6	<i>FRAME.SRT, FRAME(x).SRT</i> .....	26
9.8.7	<i>FRAME.THM, FRAME(x).THM</i> .....	26
9.8.8	<i>FRAME.TIT, FRAME(x).TIT</i> .....	26
9.8.9	<i>FRAME.LEADIN, FRAME(x).LEADIN</i> .....	26
9.8.10	<i>FRAME.TXT, FRAME(x).TXT</i> .....	26
9.8.11	<i>FRAME.SIG.XXX, FRAME(x).SIG.XXX, FRAME(x).SIG(y).XXX</i> .....	26
9.9	Variables en Dur spéciales conditionnelles.....	27
9.9.1	<i>IF.SIG(x)</i> .....	27
9.9.2	<i>IF.IMG(x)</i> .....	27
9.9.3	<i>IF.FILE(x)</i> .....	27

9.9.4	<i>IF.LNK(x)</i> .....	27
9.9.5	<i>IF.FRAME(x)</i> .....	27
9.10	Boucle de sommaire : variable de « de Pattern ».....	28
9.10.1	<i>Intérêt</i> .....	28
9.10.2	<i>Syntaxe</i> .....	28
9.10.3	<i>Exemple pour une rubrique</i> .....	28
9.10.4	<i>Exemple pour un article</i> .....	29
9.10.5	<i>Imbrication de pattern</i> .....	29
9.10.6	<i>Variables associées : les compteurs</i> .....	30
9.11	Fonctions RexPublisher.....	31
9.11.1	<i>ALPHAWIDTH</i> .....	31
9.11.2	<i>AND</i> .....	31
9.11.3	<i>ART.INDEX</i> .....	31
9.11.4	<i>ARTNBR</i> .....	31
9.11.5	<i>ART.POS</i> .....	31
9.11.6	<i>ASCII</i> .....	31
9.11.7	<i>ASKUSER</i> .....	32
9.11.8	<i>BREAK.ON</i> .....	32
9.11.9	<i>BREAK.OFF</i> .....	32
9.11.10	<i>CREATEPDF</i> .....	32
9.11.11	<i>CHR</i> .....	32
9.11.12	<i>CURR_RUBNBR</i> .....	32
9.11.13	<i>CURR_SRBNBR</i> .....	33
9.11.14	<i>CURR_SSRBNBR</i> .....	33
9.11.15	<i>CREATEPDF</i> .....	33
9.11.16	<i>DATEADD</i> .....	33
9.11.17	<i>DAY</i> .....	34
9.11.18	<i>DAYNAME</i> .....	34
9.11.19	<i>DIFFERENT</i> .....	34
9.11.20	<i>DIVIDE</i> .....	34
9.11.21	<i>DOC_INSERTTOC</i> .....	34
9.11.22	<i>EQUAL</i> .....	34
9.11.23	<i>EVEN</i> .....	34
9.11.24	<i>EXISTS</i> .....	34
9.11.25	<i>FILE_EXTENTION</i> .....	34
9.11.26	<i>FILESIZE</i> .....	34
9.11.27	<i>FILEMODIFIED</i> .....	35

9.11.28	<i>FILEPATH</i> .....	35
9.11.29	<i>FLOOR</i> .....	35
9.11.30	<i>FORMAT</i> .....	35
9.11.31	<i>GETSLICE</i> .....	35
9.11.32	<i>GENPATH</i> .....	36
9.11.33	<i>GETVAR</i> .....	36
9.11.34	<i>IMGPATH</i> .....	36
9.11.35	<i>INSERTEDIMGPATH</i> .....	36
9.11.36	<i>INSERTPDF</i> .....	36
9.11.37	<i>LOCATION</i> .....	36
9.11.38	<i>LENGTH</i> .....	36
9.11.39	<i>LESS</i> .....	36
9.11.40	<i>LIKE</i> .....	36
9.11.41	<i>LOOP</i> .....	37
9.11.42	<i>LOWER</i> .....	38
9.11.43	<i>MINUS</i> .....	38
9.11.44	<i>MODULO</i> .....	38
9.11.45	<i>MONTH</i> .....	38
9.11.46	<i>MONTHNAME</i> .....	38
9.11.47	<i>MORE</i> .....	38
9.11.48	<i>MOVEFILE</i> .....	38
9.11.49	<i>MULTIPLY</i> .....	38
9.11.50	<i>NOBLANK</i> .....	39
9.11.51	<i>NODEINDEX</i> .....	39
9.11.52	<i>NODENAME</i> .....	39
9.11.53	<i>ODD</i> .....	39
9.11.54	<i>OR</i> .....	39
9.11.55	<i>PLUS</i> .....	39
9.11.56	<i>PS_ACTIONS</i> .....	39
9.11.57	<i>PROCESSXLS</i> .....	40
9.11.58	<i>PUBPATH</i> .....	40
9.11.59	<i>READFILE</i> .....	40
9.11.60	<i>READPASTESAVE</i> .....	40
9.11.61	<i>RELATIVEPATH</i> .....	40
9.11.62	<i>REMOVEFILE</i> .....	40
9.11.63	<i>REPLACE</i> .....	40
9.11.64	<i>REPLACEBLANK</i> .....	40

9.11.65	RESIZE.....	40
9.11.66	ROUND.....	41
9.11.67	RUBNBR.....	41
9.11.68	SETVAR.....	41
9.11.69	SRBNBR.....	41
9.11.70	SSRBNBR.....	41
9.11.71	SUBSTR.....	41
9.11.72	THIS.....	42
9.11.73	TODAYI.....	42
9.11.74	TRANSCODE.....	42
9.11.75	TRANSCODE.OFF.....	42
9.11.76	TRANSCODE.ON.....	42
9.11.77	TRIM.....	42
9.11.78	UNWRAP.....	42
9.11.79	UPPER.....	42
9.11.80	URLENCODE.....	42
9.11.81	VALIDFILENAME.....	42
9.11.82	VALIDFILEPATH.....	43
9.11.83	WRITE_IF.....	43
9.11.84	WRITE_FILE.....	43
9.11.85	YEAR.....	43
9.12	Fonction d'Accès à la base de données de la Publication.....	44
9.12.1	Fonctions d'accès.....	44
9.12.2	Variables d'accès.....	45
9.13	Accès à d'autres bases de données.....	47
9.13.1	EXOSQL.....	47
9.13.2	Variables d'accès aux BD externes.....	47
9.14	Accès au contenu d'une feuille Excel.....	48
9.14.1	Fonction d'appel ; PROCESSXLS.....	48
9.14.2	Balise <!/@TABLE.BEGIN>.....	48
9.14.3	Balise <!/@TABLE.END>.....	48
9.14.4	Balise <!/@TABLE.ROW.BEGIN >.....	48
9.14.5	Balise <!/@TABLE.ROW.END>.....	49
9.14.6	Balise <!/@TABLE.CELL(X,Y)>.....	49
9.14.7	Balise <!/@TABLE.VOID(x,y)>.....	49



## 0 Préambule

Le présent manuel a pour but d'expliquer la démarche de conception et de réalisation de feuilles de styles destinées au logiciel RexPublisher™.

Il est évidemment souhaitable d'avoir une bonne compréhension :

- De RexPublisher : on se référera utilement au manuel d'utilisation du logiciel téléchargeable sur le site ([www.rexpublisher.com](http://www.rexpublisher.com)).
- Des langages cibles : HTML, XML, .... Rappelons en effet que l'on n'est limité à la génération de pages HTML, RexPublisher permet la création de tout document de formats descriptibles en ASCII.

Il est donc plus particulièrement destiné aux webmestres chargés de la création de feuilles ou aux concepteurs de templates désireux de connaître les contraintes éventuelles induites par leur mise en oeuvre.

## 1 Qu'est-ce qu'une feuille de style ?

Une feuille de style est un fichier à partir duquel RexPublisher va pouvoir générer l'ensemble des documents relatifs à un certain contenu dans une certaine forme.

Par exemple,

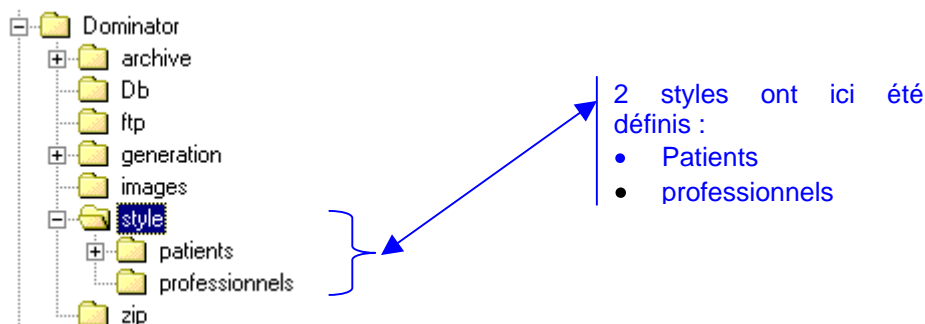
- les pages d'un site dans un format HTML
- une base XML
- Des fichiers TXT ou RTF,
- Etc..

Plus précisément, Pour un type de contenu donné, une feuille de style contient un ensemble de règles permettant de déterminer

- Dans quelle forme sera publié ce contenu
- Dans quel format (htm, rtf, txt, etc..) sera publié ce contenu
- A quel endroit sera sauvegardé ce contenu
- A quel endroit sera publié ce contenu (dans le cas de la publication sur site)

## 2 Où sont rangées les feuilles de style ?

Pour une publication donnée, et pour un style donné (eg. `Patients`) les feuilles de styles sont stockées dans un répertoire `Patients` se trouvant dans le répertoire `Style` de la publication.



## 3 Feuille de style et génération

Les feuilles de styles ont pour objet de générer des documents. Pour ce qui concerne par exemple le cas de documents HTML et plus exactement d'un site Internet, les documents générés sont placés dans un sous-répertoire du répertoire « génération », qui porte habituellement le même nom que le style qui les a générés. Ces pages internet peuvent faire référence à des fichiers de la charte graphique du site : Images, scripts, css, etc..

ces fichiers doivent être présents, à plat ou dans des sous-répertoires dédiés, dans le répertoire de destination du site sur le disque local. Le site distant devra évidemment comporter les mêmes répertoires et images

## 4 Quels sont les différents type de styles ?

RexPublisher pouvant générer plusieurs types de contenus :

- Agenda : gestion d'événements
- News : gestion d'articles

Les feuilles de styles sont « typées » pour adresser spécifiquement un type de contenu donné. Ce typage s'exprime simplement dans un **préfixe** présent dans le nom du fichier contenant la feuille de style :

- **Agenda** : Agenda.XXXX.sty
- **News** : News.XXXX.sty

Où

- **Agenda, News** sont les préfixes caractérisant le type de style
- **XXXX** une chaîne de caractères précisant sur quel sous-contenu porte la feuille de style
- **Sty** est l'extension d'une feuille de style RexPublisher

## 5 Quels sont les documents publiables ?

Pour un style donné, les documents générables et publiables par RexPublisher sont structurellement les suivants :

1. Le sommaire général (*Sommaire*)
2. Les sommaires de Section
3. Les sommaires de Rubriques
4. Les sommaires de sous-rubriques
5. Les sommaires par Thèmes
6. Les sommaires par mots-clés
7. Les sommaires par date (pour le type **Agenda**)
8. Les Articles unitaires

Par ailleurs, une feuille de style neutre (c'est à dire ne donnant lieu à aucune génération de fichier) est dédiée à la définition des paramètres de publication;

## 6 Comment nommer les feuilles de style ?

Comme vu au point 2, une feuille de style est déjà caractérisée par :

- **Un Préfixe** : caractérisant le type de contenu sur lequel porte la feuille de style (agenda, news, links, others)
- **Une extension .sty** : caractérisant les feuilles de style RexPublisher

De plus, chaque type de feuille de style, suivant le document cible, doit être nommé de manière spécifique (on utilisera ici le préfixe **News** pour exemple) :

*Remarque* : l'ordre dans lequel les options de nommages sont donnée correspond à l'ordre dans lequel RexPublisher recherche les feuilles de style applicables.

### 6.1 Les paramètres de publication

Les paramètres de publication sont dans une feuille de style invariablement nommée :

publication.sty

*Remarque* : Ces paramètres sont donc définis pour tout type de données traitées par le style ( Newsn Agenda, et Others)

### 6.2 Le sommaire général :

Le sommaire général doit être invariablement nommé :

News.sommaire.sty

(ou agenda.sommaire.sty pour agenda, ou others.sommaire.sty pour Others)

### 6.3 Les sommaires de Sections

Pour une section nommée « **MaSection** », sa feuille de style de sommaire peut être nommée de 2 manières

1. Pour une feuille de style spécifique à cette section :

`News.MaSection.sty`

2. Pour une feuille de style générique à toutes les sections :

`News.default.sty`

### 6.4 Les sommaires de Rubrique

Pour une rubrique nommée « **MaRubrique** » appartenant à la section nommée « **MaSection** », sa feuille de style de sommaire peuvent être nommé de plusieurs manières

1. Pour une feuille de style spécifique à cette rubrique dans cette section :

`News.MaSection.MaRubrique.sty`

2. Pour une feuille de style générique à toutes les rubrique de cette section :

`News.MaSection.default.sty`

3. Pour une feuille de style générique à toutes les rubriques de toutes sections

`News.default.default.sty`

### 6.5 Les sommaires de sous-Rubriques

Pour une sous-rubrique nommée « **MaSous-Rubrique** » appartenant à la rubrique nommée « **MaRubrique** » appartenant elle-même à la section nommée « **MaSection** », sa feuille de style de sommaire peut être nommée de plusieurs manières

1. Pour une feuille de style spécifique à cette sous-rubrique dans cette rubrique dans cette section :

`News.MaSection.MaRubrique.MaSousRubrique.sty`

2. Pour une feuille de style générique à toutes les sous-rubriques de cette rubrique de cette section :

`News.MaSection.MaRubrique.default.sty`

3. Pour une feuille de style générique à toutes les sous-rubrique de toutes les rubriques de cette section :

`News.MaSection default.default.sty`

4. Pour une feuille de style générique à toutes les sous-rubrique de toutes les rubriques de toutes sections

`News.default default.default.sty`

### 6.6 Les sommaires de Themes

Pour un thème nommé « **MonThème** », la feuille de style associée peut être nommée de plusieurs manières :

1. Pour une feuille de style spécifique à ce thème :

`News.theme.MonThème.sty`

2. Pour une feuille de style générique pour tous les thèmes :

`News.theme.default.sty`

## 6.7 Les sommaires de Mots-clé

Pour un mot-clé nommé « **MonMotClé** », la feuille de style associée peut être nommée de plusieurs manières :

3. Pour une feuille de style spécifique à ce mot-clé :

`News.keyword.MonMotClé.sty`

4. Pour une feuille de style générique pour tous les mots-clés :

`News.keyword.default.sty`

## 6.8 Les sommaires d'articles

Pour un article appartenant à la sous-rubrique « **MaSous-Rubrique** » appartenant à la rubrique nommée « **MaRubrique** » appartenant elle-même à la section nommée « **MaSection** », sa feuille de style de sommaire peut être nommée de plusieurs manières

1. Pour une feuille de style spécifique à cette sous-rubrique dans cette rubrique dans cette section :

`News.art.MaSection.MaRubrique.MaSousRubrique.sty`

2. Pour une feuille de style générique à toutes les sous-rubriques de cette rubrique de cette section :

`News.art.MaSection.MaRubrique.default.sty`

Ou

`News.art.MaSection.MaRubrique.sty`

3. Pour une feuille de style générique à toutes les sous-rubrique de toutes les rubriques de cette section :

`News.MaSection.default.default.sty`

Ou

`News.art.MaSection.default.sty`

Ou

`News.art.MaSection.sty`

4. Pour une feuille de style générique à toutes les sous-rubrique de toutes les rubriques de toutes sections :

`News.art.default.default.default.sty`

Ou

`News.art.default.sty`

## 7 Comment est structurée une feuille de style

### 7.1 Qu'est ce qu'une Balise RexPublisher

Une feuille de style est composée d'une suite de différentes **Balises RexPublisher**. Une balise RexPublisher est toujours composée de la manière suivante :

```
< !@MABALISE>  ← Balise ouvrante
    corps de la balise
< ! /@MABALISE> ← Balise fermante (noter le « / » après le « ! »)
```

### 7.2 Les Sous-balises RexPublisher

Une balise RexPublisher peut en fait être vue comme le lieu de définition d'une variable nommée MABALISE.

La valeur de cette variable peut être définie en fonction de l'appartenance du contenu manipulé avec la feuille de style. Il est en effet possible de spécialiser la valeur de la variable suivant sont appartenance à :

- Une Section donnée
- Une Rubrique donnée
- Une Sous-rubrique donnée

C'est dans le corps de la balise que cette spécialisation est décrite, à l'aide de sous balise RexPublisher de la forme suivante :

#### 7.2.1 Spécialisation par Section

- Valeur pour la Section « **MaSection** »

```
< !@MaSection >
    corps de la balise
< ! /@MaSection >
```

- Valeur pour toute Section

```
< !@DEFAULT>
    corps de la balise
< ! /@DEFAULT>
```

#### 7.2.2 Spécialisation par Section et Rubrique

- Valeur pour la Section « **MaSection** » et une rubrique « **MaRubrique** »

```
< !@MaSection .MaRubrique >
    corps de la balise
< ! /@MaSection .MaRubrique >
```

- Valeur pour toute Rubrique de la Section « **Ma Section** »

```
< !@MaSection .DEFAULT>
    corps de la balise
< ! /@ MaSection .DEFAULT>
```

- Valeur pour toute Rubrique de toute Section

```
< !@DEFAULT .DEFAULT>
    corps de la balise
< ! /@DEFAULT .DEFAULT>
```

### 7.2.3 Spécialisation par Section, Rubrique, et Sous-Rubrique

- Valeur pour la sous-Rubrique « **MaSousRubrique** » de la rubrique « **MaRubrique** » de la Section « **MaSection** »
 

```
<!@MaSection . MaRubrique . MaSousRubrique >
      corps de la balise
      <!/@MaSection . MaRubrique . MaSousRubrique>
```
- Valeur pour toute sous-Rubrique de la Rubrique « **MaRubrique** » de la Section « **MaSection** »
 

```
<!@MaSection . MaRubrique . DEFAULT>
      corps de la balise
      <!/@MaSection . MaRubrique . DEFAULT>
```
- Valeur pour toute sous-rubrique de toute Rubrique de la Section « **MaSection** »
 

```
<!@MaSection DEFAULT . DEFAULT>
      corps de la balise
      <!/@MaSection . DEFAULT . DEFAULT>
```
- Valeur pour toute sous-rubrique de toute Rubrique de toute Section
 

```
<!@DEFAULT . DEFAULT . DEFAULT>
      corps de la balise
      <!/@DEFAULT . DEFAULT . DEFAULT>
```

### 7.2.4 Quelques finesses sur les corps de balises

Il est parfois utiles, voire nécessaire – de spécialiser les corps de balise en caractérisant la section, rubrique, ou sous-rubrique ou article **en cours** de manière explicite. Par en cours, on signifie la section, rubrique, sous-rubrique, ou article pour lequel la feuille de style a été appelée. Ceci est particulièrement vrai dans les feuilles de style générique (défaut).

#### 7.2.4.1 Exemple pour une section, rubrique, ou sous-rubrique (MYSELF)

Dans ce cas, on peut caractériser la valeur associé à cette section, rubrique ou sous rubrique dans le corps d'une balise en remplaçant le nom de la section, rubrique, ou sous-rubrique par **MYSELF**.

```
<!@MYSELF . DEFAULT . DEFAULT>
      corps de la balise
      <!/@MYSELF . DEFAULT . DEFAULT>
```

#### 7.2.4.2 Exemple pour un ARTICLE

Dans le cas d'un article, on peut caractériser la valeur associée à lui même, sa section, rubrique ou sous rubrique dans le corps d'une balise en remplaçant le nom de la section, rubrique, ET sous-rubrique par **ME**.

```
<!@ME>
      corps de la balise
      <!/@ME>
```

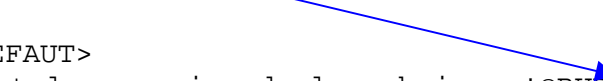
### 7.3 Corps de sous-balises

Le corps des sous-balises RexPublisher contiennent les valeurs finales des variables utilisateur.

Cette valeur est en fait constitué de texte (html, rtf, txt, etc..) dans lequel sont éventuellement présente des **variables RexPublisher**.

Exemple :

```
<!--MABALISE>
<!--MASECTION.DEFAULT.DEFAULT>
    Bonjour ! ceci est le sommaire de la rubrique !@RUB@!
<!--/MASECTION.DEFAULT.DEFAULT>
<!--/MABALISE>
```



L'appel à une variable RexPublisher dans le texte d'un corps de balise est toujours de la forme :

!@MAVARIABLE@! ← Noter le !@ ouvrant et le @! fermant

### 7.4 Variables RexPublisher

#### 7.4.1 Types de variables

Les variables RexPublisher sont de plusieurs types :

1. **Variables en dur** : Il s'agit de variable dont le nom, connu de RexPublisher, fait référence au contenu des données manipulées par RexPublisher (eg. Un titre, le nom d'une section, la valeur du champ texte, etc..)
2. **Variables Utilisateur** : Il s'agit soit :
  - de spécialisation de variable en dur,
  - de variables nommées arbitrairement par l'utilisateur.
3. **Fonctions** : Il s'agit de variables dont le nom, connu de RexPublisher, fait référence à des fonctions câblées du logiciel (SUBSTR, DATE, etc...)

#### 7.4.2 Syntaxe

Une variable RexPublisher peut prendre les formes suivantes :

1. **Variable simple** : !@MAVARIABLE@! : où MAVARIABLE est une chaîne de caractères
2. **Variable simple conditionnelle** : !@IF.MAVARIABLE@! : où MAVARIABLE est une chaîne de caractères.
3. **Variable développée** : !@MAVARIABLE(Expression)@! : où expression est une chaîne de caractères comprenant éventuellement des variables RexPublisher.

exemple : !@TRAITER(!@MADATE@!)@!

Note : expression sera toujours évaluée avant que ne soit évaluée la variable elle-même.

4. **Variable développée conditionnelle** : !@IF.MAVARIABLE(Expression)@! : où expression est une chaîne de caractères comprenant éventuellement des variables RexPublisher.

exemple : !@IF.RUBRIQUE(!@RUB@!)@!

Note : expression sera toujours évaluée avant que ne soit évaluée la variable elle-même.



5. **Variable développée avec suffixe** : `!@MAVARIABLE(Expression).suffixe@!` :

où :

- `expression` est une chaîne de caractères comprenant éventuellement des variables RexPublisher.
- `Suffixe` : est une chaîne de caractères « neutre » i.e. un simple identifiant

exemple : `!@TRAITER(!@MADATE@!).jour@!`

Note : `expression` sera toujours évaluée avant que ne soit évaluée la variable elle-même.

6. **Variable développée conditionnelle avec suffixe** :

`!@IF.MAVARIABLE(Expression).suffixe@!` : où :

- `expression` est une chaîne de caractères comprenant éventuellement des variables RexPublisher.
- `Suffixe` : est une chaîne de caractères « neutre » i.e. un simple identifiant

exemple : `!@IF.RUBRIQUE(!@RUB@!).Section@!`

Note : `expression` sera toujours évaluée avant que ne soit évaluée la variable elle-même.

## 7.5 Balises Obligatoires

Un certain nombre de balises doivent toujours être présentes dans une feuille de style.

### 7.5.1 Balise `<!@DOC>`

Cette balise va permettre de définir le corps principal du texte devant être généré.

*Remarque* : la balise `!@DOC` **est la seule** à ne pas avoir de corps de balise spécialisé par Section, Rubrique et sous Rubrique.

### 7.5.2 Balise `<!@NAME>`

Cette balise va permettre de définir nom du fichier généré, *hors extension*.

### 7.5.3 Balise `<!@EXTENTION>`

Cette balise va permettre de définir l'extension du nom du fichier généré.

### 7.5.4 Balise `<!@PATH>`

Cette balise va permettre de définir le chemin du répertoire où sera copié le fichier généré.

## 7.6 Balises Optionnelles

Des balises sont optionnelles ou conditionnées à la présence d'images ou de fichiers :

### 7.6.1 Balise `<!@IMGPATH>`

Cette balise va permettre de définir le chemin du répertoire où seront copiées les images associées au document généré à partir de la feuille de style.

### 7.6.2 Balise `<!@FILEPATH>`

Cette balise va permettre de définir le chemin du répertoire où seront copiées les fichiers associés au document généré à partir de la feuille de style.

### 7.6.3 Balise `<!@INSERTEDIMGPATH>`

Cette balise va permettre de définir le chemin du répertoire où seront copiées les images incluses dans le texte enrichissable de l'article généré à partir de la feuille de style.

### 7.6.4 Balise `<!@BRKSTY>`

Cette balise va permettre de définir la manière dont sera codé le retour à la ligne.

### 7.6.5 Balise `<!@HREFPAT(Type)>`

Cette balise, utilisée en cas de publication HTML, va permettre de définir la manière dont seront codés les liens hypertextes posés sur le texte d'un article, en fonction des types qui y auront été associés par l'éditeur du contenu. Si aucun style n'est associé au type le lien hyper-texte reste inchangé :

Exemple :

```
<!@HREFPAT(Popup1)>  
<!@DEFAULT>javascript:MM_openBrWindow('!@@!', '', 'scrollbars=yes,resizable=yes,width=450,height=500')</@DEFAULT>  
</@HREFPAT(Popup1)>
```

définit le style associé au type `Popup1`, où `!@@!` est la variable générique qui sera remplacé par l'URL du lien.

### 7.6.6 Balise `<!@ HIDEHREF>`

Cette balise, utilisée surtout en cas d'utilisation multi-utilisateur de Rexpublisher V1, va permettre de cacher, si nécessaire, de rendre les liens hypertextes générés par Rexpublisher sur les documents prévisualisés insensibles.

Il suffit pour cela que la valeur de la balise soit égale à « True »

## 7.7 Balises de fichiers associés

Certaines balises définissent des relations spécifiques entre la feuille de style et d'autres fichiers style associés.

### 7.7.1 Balise `<!@INCLUDE>`

Cette balise va permettre de d'inclure dans la feuille de style en cours le contenu de fichiers, dont les chemins, relatif à la feuille de style en cours, sont simplement listés entre les 2 balises.

```
<!@INCLUDE>
../toto/moninclude.sty
../titi/unautre.sty
<!@INCLUDE>
```

Ce qui peut être bien utile pour partager des définitions identiques sans avoir à les répliquer.

### 7.7.2 Balise `<!@ATTACHED TO>`

Cette balise va permettre de déclencher automatiquement l'évaluation de feuille de style associées à la feuille de style courante, de manière transparente à l'utilisateur. Les chemins de ces feuilles, relatif à la feuille de style en cours, sont simplement listés entre les 2 balises.

```
<!@ATTACHED TO>
monjavascript.sty
monindex.sty
<!@ATTACHED TO>
```

## 7.8 Balises de Publication

La feuille de style `publication.sty` contient toujours les paramètres de publication finale.

Ils sont définis dans une balise spéciale :

### 7.8.1 Balise `<!@PUBLICATION>`

La Balise `PUBLICATION` est constituée de plusieurs sous balises définissant les paramètres de publication. Elle décrit à la fois le mode de publication et éventuellement l'adresse de Publication distante des pages associées au style.

#### 7.8.1.1 Sous-Balise `TYPE`

La sous-balise `TYPE` décrit le mode de publication des pages associées au style. Elle peut avoir les valeurs suivantes :

- `FTP` : publication par FTP (option 1)
- `FTP2` : publication par FTP (option 2)
- `SFTP` : publication SFTP
- `MAIL` : publication par MAIL
- `COPY` : publication par simple copie

- ZIP : publication par ZIP : les pages générées seront regroupées dans un fichier sauvegardés dans le répertoire **zip** de la publication et nommé : **N°édition\_Date\_édition.zip** où N°édition et Date\_édition sont les N° d'édition et la date de l'édition en cours de traitement.

#### 7.8.1.2 *Sous-Balise ADDRESS*

La sous-balise ADDRESS décrit éventuellement l'adresse de Publication distante des pages associées au style.

Elle a de sens que pour les publications FTP, FTP2, SFTP, MAIL et COPY

#### 7.8.1.3 *Sous-Balise ROOT*

cette sous-balise n'est significative que dans le cas d'une publication FTP. Elle décrit le répertoire racine de la publication distante.

#### 7.8.1.4 *Sous-Balise MODE*

Cette sous-balise n'est significative que dans le cas d'une publication FTP. Elle décrit le mode de dialogue avec le serveur. 2 valeurs sont possibles :

- ACTIVE : valeur par défaut
- PASSIVE

#### 7.8.1.5 *Sous-Balise USER*

Cette sous-balise n'est significative que dans le cas d'une publication FTP.

Elle décrit le Login de connexion FTP. Elle peut ne pas être instruite ; dans ce cas, au moment de la publication proprement dite, RexPublisher demandera à l'utilisateur de la saisir.

#### 7.8.1.6 *Sous-Balise PASSWORD*

Cette sous-balise n'est significative que dans le cas d'une publication FTP.

Elle contient le Password de connexion FTP. Elle peut ne pas être instruite ; dans ce cas, au moment de la publication proprement dite, RexPublisher demandera à l'utilisateur de la saisir.

#### 7.8.1.7 *Sous-Balise LROOT*

Cette sous-balise va décrire le répertoire racine sous lequel les pages générées vont être localement copiées. Si cette balise n'est pas renseignée, le répertoire racine est du nom du style..

#### 7.8.1.8 *Sous-Balise WEB*

Cette sous-balise va contenir l'adresse URL du site web final.

#### 7.8.1.9 *Sous-Balise WORKFLOWSERVER*

Cette sous-balise va contenir l'adresse SMTP du serveur de mail utilisé tant pour la publication MAIL que pour la gestion des notifications dans le cadre du workflow.

### 0.1.1 Exemples :

```
< !@PUBLICATION>
  < !@TYPE>SCP< !/@TYPE>
  < !@MODE>ACTIVE< !/@MODE>
```

```
<!--ADDRESS>66.349.1367.138<!--ADDRESS>
<!--ROOT>essaid<!--ROOT>
<!--LROOT>htm<!--LROOT>
<!--USER>cervonic<!--USER>
<!--PASSWORD>toto.com<!--WEB>
<!--WORKFLOWSERVER><!--WORKFLOWSERVER>
<!--PUBLICATION>
ou bien encore

<!--PUBLICATION>
<!--ADDRESS>zip<!--ADDRESS>
<!--ROOT><!--ROOT>
<!--USER><!--USER>
<!--PASSWORD><!--PASSWORD>
<!--LROOT>site<!--LROOT>
<!--WEB>http://www.monsite.com<!--WEB>
<!--PUBLICATION>
```

## 8 Comment est interprétée une feuille de style

### 8.1 Recherche de la feuille de style

Pour un style donnée, un type de données donné (ici `News`), et suivant la nature du document que l'on veut générer, RexPublisher effectue la recherche suivante (en supposant que les conditions sont les suivantes) :

- Section : `MaSection`
- Rubrique : `MaRubrique`
- Sous-Rubrique : `MaSousRubrique`
- Theme : `MonTheme`
- Mot-Clé : `MonMotClé`

#### 8.1.1 Le sommaire général

La feuille de style recherchée est :

```
news.sommaire.sty
```

#### 8.1.2 Les sommaires de Section

RexPublisher cherche d'abord la feuille de style spécifique à la section :

```
news.masection.sty
```

et, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.default.sty
```

#### 8.1.3 Les sommaires de Rubriques

RexPublisher cherche d'abord la feuille de style spécifique à la rubrique :

```
news.masection.marubrique.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.masection.default.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.default.default.sty
```

#### 8.1.4 Les sommaires de sous-rubriques

RexPublisher cherche d'abord la feuille de style spécifique à la sous-rubrique :

```
news.masection.marubrique.masousrubrique.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.masection.marubrique.default.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.Masection.default.default.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.default.default.default.sty
```

### 8.1.5 Les Articles unitaires

RexPublisher cherche d'abord la feuille de style spécifique à la sous-rubrique :

```
News.art.masection.marubrique.masousrubrique.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.art.masection.marubrique.default.sty
```

ou

```
news.art.masection.marubrique.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.art.Masection.default.default.sty
```

ou

```
news.art.Masection.default.sty
```

ou

```
news.art.Masection.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.art.default.default.default.sty
```

ou

```
news.art.default.default.sty
```

ou

```
news.art.default.sty
```

### 8.1.6 Les sommaires par Thèmes

RexPublisher cherche d'abord la feuille de style spécifique au mot-clé:

```
news.theme.montheme.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.theme.default.sty
```

### 8.1.7 Les sommaires par Mot-clé

RexPublisher cherche d'abord la feuille de style spécifique au mot-clé :

```
news.keyword.monmotclé.sty
```

puis, au cas où la précédente n'existerait pas, RexPublisher recherche :

```
news.keyword.default.sty
```

## 8.2 *Écriture du document*

### 8.2.1 Constitution du fichier

Une fois la feuille de style trouvée, RexPublisher va construire le nom du fichier à générer :

- A l'aide de la balise !@NAME@ (pour le corps du nom)
- A l'aide de la balise !@EXTENSION@ (pour l'extension du fichier)
- A l'aide de la balise !@PATH@ (pour le chemin de sauvegarde du fichier)

*Remarque* : RexPublisher va se charger de « mettre d'équerre » les nom et chemins de fichiers. Il remplacera ainsi tous caractères illégaux par un « \_ » (underscore ou blanc souligné en français)

### 8.2.2 Recherche de la balise !@DOC

RexPublisher va ensuite chercher la valeur (i.e. le corps de balise) de la balise !@DOC, qui constituera le corps du texte du document.

RexPublisher va commencer à écrire le corps du texte dans le fichier nommé et positionné d'après la phase précédente, en appliquant récursivement la règle suivante :

**Règle** : à chaque fois que RexPublisher va rencontrer une variable RexPublisher dans la chaîne qu'il est en train d'écrire, il essaye de lui substituer sa valeur.

### 8.2.3 Recherche de la valeur d'une variable

Quand RexPublisher cherche à substituer une variable par sa valeur il applique récursivement la règle suivante :

#### 1. **La variable est elle une variable « en Dur » :**

*Exemple* : (eg. !@RUB@ !.) :

Dans ce cas, il lui substitue simplement sa valeur

#### 2. **La variable est elle une variable « en Dur » conditionnelle :**

*Exemple* : (eg. !@IF.RUB@!, !@IF.TIT@!, !@IF.TXT@!, etc.) ?

Dans ce cas, et seulement si la variable en dur n'est pas vide, RexPublisher va chercher dans la feuille de style la définition de la variable conditionnelle pour lui substituer sa valeur.

#### 3. **La variable est elle une variable « en Dur » conditionnelle développée avec ou sans suffixe :**

*Exemple* : (eg. !@IF.RUB(expression).SUFFIXE@!, etc.) ?

Dans ce cas, et seulement si la variable en dur n'est pas vide, RexPublisher va chercher dans la feuille de style la définition de la variable conditionnelle pour lui substituer sa valeur.

**Note** : si expression est « ? », ce « ? » est évalué en la valeur de la variable en dur (eg. La valeur du Titre pour !@IF.TIT(?)@!)



**4. La variable est elle une variable développée, avec ou sans suffixe :**

Exemple : (eg. !@MAVARIABLE(expression).SUFFIXE@!)

Dans ce cas, RexPublisher va :

- Evaluer d'abord l'expression
- Rechercher la définition de la variable !@MAVARIABLE(expression évaluée).SUFFIXE@!, et si elle existe lui substituer sa valeur
- Sinon Rechercher la définition de la variable !@MAVARIABLE(?).SUFFIXE@!, et si elle existe lui substituer sa valeur

**5. La variable est un appel de fonction, développée ou non, avec ou, sans suffixe :**

Exemple : (eg. !@UPPER@!)

Dans ce cas, RexPublisher évalue un à un ses arguments, puis appelle la fonction sur ces arguments instanciés, et lui substitue simplement la valeur retournée.

## 9 Liste des Variables et fonctions RexPublisher

### 9.1 Variables en Dur

#### 9.1.1 PUBLICATION

Contient le nom de la PUBLICATION courante : eg. *Egora*

#### 9.1.2 EDITION.NB

Contient le N° D'EDITION courante

#### 9.1.3 EDITION.DATE ou DATE

Contient la DATE de l'édition courante

#### 9.1.4 RUB

Contient le nom de la SECTION courante

#### 9.1.5 SRB

Contient le nom de la RUBRIQUE courante

#### 9.1.6 SSRB

Contient le nom de la SOUS-RUBRIQUE courante

#### 9.1.7 SRT

Contient le SUR-TITRE de l'article en cours.

#### 9.1.8 THM

Contient le THEME de l'article en cours.

#### 9.1.9 TIT

Contient le TITRE de l'article en cours.

#### 9.1.10 CHA

Contient le CHAPEAU de l'article en cours.

#### 9.1.11 TXT

Contient le TXT de l'article en cours.

#### 9.1.12 SIG

Contient la signature de l'article en cours.

#### 9.1.13 KWDLIST

Contient la liste des mots-clés de l'article en cours.

### 9.2 Variables en Dur conditionnelle

Les variables en dur conditionnelles correspondent très exactement aux variables en dur :

- Précédée de IF. (eg. IF.RUB)
- Ou Suffixé par STY (eg. RUBSTY)

### 9.3 Variables en Dur spéciales

Certaines variables en dur font référence à des caractéristiques du contenu de l'édition.

### 9.3.1 RUB.IDX

Contient le N° d'index de la SECTION en cours de manipulation dans la liste des rubriques connues de RexPublisher (commence à 1).

Note : Dans une édition RexPublisher, certaines rubriques peuvent n'avoir aucun article. Elle n'en sont pas moins connues de RexPublisher.

Ainsi, si dans une édition RexPublisher, on recense 3 rubriques (Toto, Titi, Tata), les index des rubriques sont (qu'elles contiennent ou non des articles) :

- Toto : 1
- Titi : 2
- Tata : 3

### 9.3.2 CURR RUB.IDX

Contient le N° d'index de la SECTION de l'article en cours de manipulation dans la liste des rubriques connues de RexPublisher (commence à 1).

Note : La différence entre CURR\_RUB.IDX et RUB.IDX est que la première ne varie pas pendant toute l'exploitation d'une feuille de style. Elle fait référence au contexte d'appel de la feuille de style (qui se réfère toujours à une section, rubrique et sous-rubrique).

### 9.3.3 RUB.ORDER

A la différence de la précédent Contient le N° d'ordre de la SECTION en cours de manipulation dans la liste des rubriques connues de RexPublisher **contenant des articles**.

Ainsi, si dans une édition RexPublisher, on recense 3 sections (Toto, Titi, Tata) et **où Titi ne contient pas d'article**, les index des rubriques sont :

- Toto : 1
- Titi : -1
- Tata : 2

### 9.3.4 SRB.IDX

Contient le N° d'index de la RUBRIQUE courante dans la liste des rubriques connues de RexPublisher dans la Section courante (commence à 0).

Note : Dans une édition RexPublisher, certaines rubriques peuvent n'avoir aucun article. Elle n'en sont pas moins connues de RexPublisher. Noter aussi, qu'il existe une rubrique spéciale, dite « **VIDE** », dont l'index est toujours 0.

Ainsi, si dans une édition RexPublisher, on recense 3 rubriques (Roto, Riti, Rata), les index des rubriques sont (qu'elles contiennent ou non des articles) :

- Rubrique vide : 0
- Roto : 1
- Riti : 2
- Rata : 3

### 9.3.5 CURR SRB.IDX

Contient le N° d'index de la RUBRIQUE de l'article en cours de manipulation dans la liste des rubriques connues de RexPublisher (commence à 1).

Note : La différence entre CURR\_SRB.IDX et SRB.IDX est que la première ne varie pas pendant toute l'exploitation d'une feuille de style. Elle fait référence au contexte d'appel de la feuille de style (qui se réfère toujours à une section, rubrique et sous-rubrique).

### 9.3.6 SRB.ORDER

A la différence de la précédent Contient le N° d'ordre de la RUBRIQUE courante dans la liste des rubriques connues de RexPublisher dans la Section courante **contenant des articles**.

Ainsi, si dans une édition RexPublisher, on recense 3 rubriques (Roto, Riti, Rata) et **où Titi ne contient pas d'article**, les index des rubriques sont:

- Rubrique vide : 0
- Roto : 1
- Riti : -1
- Rata : 2

### 9.3.7 SSRB.IDX

Contient le N° d'index de la Sous-RUBRIQUE courante dans la liste des sous-rubriques connues de RexPublisher dans la rubrique courante de la section courante (commence à 0).

Note : Dans une édition RexPublisher, certaines sous-rubriques peuvent n'avoir aucun article. Elle n'en sont pas moins connues de RexPublisher. Noter aussi, qu'il existe une sous-rubrique spéciale, dite « **VIDE** », dont l'index est toujours 0.

Ainsi, si dans une édition RexPublisher, on recense 3 sous-rubriques (Foto, Fiti, Fata), les index des rubriques sont (qu'elles contiennent ou non des articles) :

- Sous-Rubrique vide : 0
- Foto : 1
- Fiti : 2
- Fata : 3

### 9.3.8 CURR\_SSRB.IDX

Contient le N° d'index de la Sous-RUBRIQUE de l'article en cours de manipulation dans la liste des rubriques connues de RexPublisher (commence à 1).

Note : La différence entre CURR\_SSRB.IDX et SSRB.IDX est que la première ne varie pas pendant toute l'exploitation d'une feuille de style. Elle fait référence au contexte d'appel de la feuille de style (qui se réfère toujours à une section, rubrique et sous-rubrique)

### 9.3.9 SSRB.ORDER

A la différence de la précédent Contient le N° d'ordre de la SOUS-RUBRIQUE courante dans la liste des sous-rubriques connues de RexPublisher dans la rubrique courante de la section courante **contenant des articles**.

Ainsi, si dans une édition RexPublisher, on recense 3 sous-rubriques (Foto, Fiti, Fata) et **où Titi ne contient pas d'article**, les index des rubriques sont:

- Sous-Rubrique vide : 0
- Foto : 1
- Fiti : -1
- Fata : 2

### 9.3.10 ART.INTNB

Contient le NUMERO interne de l'article en cours.

Note : Chaque article d'une édition à un N° interne qui ne varie pas, même quand on le change de place dans l'édition (changement de section, de rubrique, ou de sous-rubrique).

Ce numéro lui est donné une fois pour toutes à sa création. Dans une même édition, aucun article ne peut avoir le même N° interne qu'un autre

### 9.3.11 ART.LOCNB

Contient le NUMERO correspondant à la position de l'article en cours dans sa sous-rubrique (commence à 1).

### 9.3.12 ART.POS(Profondeur)

En fonction de la valeur de l'argument [profondeur](#) entre paranethèse, renvoie Contient le NUMERO correspondant à la position de l'article en cours dans :

- Profondeur = 0 : dans l'édition
- Profondeur = 1 : dans sa section
- Profondeur = 2 : dans sa rubrique
- Profondeur = 3 : dans sa sous-rubrique
- [Pas d'argument](#) : dans sa sous-rubrique

## **9.4 Variables relatives aux Signatures**

On peut faire référence à la signature en cours (ce qui correspond au cas où l'on est en train de traiter une boucle sur les signatures : cf. 8.9), ou bien à une signature en particulier, en précisant son index dans la liste des signatures associées à l'article en cours de traitement..

De manière générale, toute caractéristique de l'article pointé par le lien sont accessibles via le préfixe SIG (signature courante) ou SIG(X) (signature N°x) : de la manière suivante :

### **9.4.1 SIG.NBR**

- Contient le nombre de signatures associées à l'article en cours.

### **9.4.2 SIG.NAME, SIG(x).NAME**

- Contient le nom du signataire courant
- Contient le nom du signataire N° x

### **9.4.3 SIG.FIRSTNAME, SIG(x).FIRSTNAME**

- Contient le prénom du signataire courant
- Contient le prénom du signataire N° x

### **9.4.4 SIG.ALIAS, SIG(x).ALIAS**

- Contient le pseudo du signataire courant
- Contient le pseudo du signataire N° x

### **9.4.5 SIG.CIVILITY, SIG(x).CIVILITY**

- Contient la civilité du signataire courant
- Contient la civilité du signataire N° x

### **9.4.6 SIG.TITLE, SIG(x).TITLE**

- Contient le titre du signataire courant
- Contient le titre du signataire N° x

### **9.4.7 SIG.INITIALS, SIG(x).INITIALS**

- Contient les initiales du signataire courant
- Contient les initiales du signataire N° x

### **9.4.8 SIG.EMAIL, SIG(x).EMAIL**

- Contient l'adresse email du signataire courant
- Contient l'adresse email du signataire N° x

## **9.5 Variables relatives aux Images**

On peut faire référence à l'image en cours (ce qui correspond au cas où l'on est en train de traiter une boucle sur les images : cf. 8.9), ou bien à une image en particulier, en précisant son index dans la liste des images associées à l'article en cours de traitement.

### **9.5.1 IMG.NBR**

- Contient le nombre d'images associées à l'article en cours.

### **9.5.2 IMG.NAME, IMG(x).NAME**

- Contient le nom de l'image courante
- Contient le nom de l'image N° x

### **9.5.3 IMG.AUTHOR, IMG(X).AUTHOR**

- Contient l'auteur de l'image courante
- Contient l'auteur de l'image N° x

### **9.5.4 IMG.COPYRIGHT, IMG(x).COPYRIGHT**

- Contient le copyright de l'image courante
- Contient le copyright de l'image N° x

### **9.5.5 IMG.LINK, IMG(X).LINK**

- Contient le lien associé à l'image courante
- Contient le lien associé à l'image N° x

### **9.5.6 IMG.CAPTION, IMG(X).CAPTION**

- Contient la légende de l'image courante
- Contient la légende de l'image N° x

### **9.5.7 IMG.EXTENTION, IMG(X).EXTENTION**

Contient l'extension de l'image courante

Contient l'extension de l'image N° x

### **9.5.8 IMG.WIDTH, IMG(X).WIDTH**

- Contient la largeur en pixel de l'image courante
- renvoie la largeur de l'image N° x

### **9.5.9 IMG.HEIGHT, IMG(X).HEIGHT**

- Contient la hauteur en pixel de l'image courante
- contient la hauteur de l'image N° x

### **9.5.10 IMG.INDEX**

Contient l'index de l'image courante dans la liste des images associées à l'article en cours.

## **9.6 Variables relatives aux Fichiers**

On peut faire référence un fichier en cours (ce qui correspond au cas d'une boucle sur les fichiers : cf. 8.9), ou bien à un fichier en particulier, en précisant son index dans la liste des fichiers associés à l'article en cours de traitement.

### **9.6.1 FILE.NBR**

- Contient le nombre de fichiers associés à l'article courant

### **9.6.2 FILE.NAME, FILE(x).NAME**

- Contient le nom du fichier associé courant
- Contient le nom du fichier associé N° x

### **9.6.3 FILE.AUTHOR, FILE(X).AUTHOR**

- Contient l'auteur du fichier associé courant
- Contient l'auteur du fichier associé N° x

### **9.6.4 FILE.COPYRIGHT, FILE(x).COPYRIGHT**

- Contient le copyright du fichier associé courant
- Contient le copyright du fichier associé N° x

### **9.6.5 FILE.CAPTION, FILE(X).CAPTION**

- Contient la légende du fichier associé courant
- Contient la légende du fichier associé N° x

### **9.6.6 FILE.SIZE, FILE(X).SIZE**

- Contient la taille en KO du fichier associé courant
- Contient la taille en KO du fichier associé courant

### **9.6.7 FILE.EXTENTION, FILE(X).EXTENTION**

- Contient l'extension du fichier associé courant
- Contient l'extension du fichier associé courant

### **9.6.8 FILE.INDEX, FILE(X).INDEX**

- Contient l'index du fichier associé courant dans la liste des fichiers associés
- Contient l'index du fichier associé courant dans la liste des fichiers associés

### **9.6.9 FILE.ORIGINAL, FILE(X).ORIGINAL**

- Contient le chemin absolu du fichier associé courant
- Contient le chemin absolu du fichier associé courant



## **9.7 Variables relatives aux Liens**

On peut faire référence au lien en cours (ce qui correspond au cas où l'on est en train de traiter une boucle sur les liens : cf. 8.9), ou bien à un lien en particulier, en précisant son index dans la liste des liens associés à l'article en cours de traitement.

De manière générale, toute caractéristique de l'article pointé par le lien sont accessibles via le préfixe LNK (lien courant) ou LNK(X) (lien N°x) : de la manière suivante :

### **9.7.1 LNK.NBR**

Contient le nombre de liens associés à l'article en cours.

### **9.7.2 LNK.RUB, LNK(x).RUB**

Contient le nom de la SECTION du lien N° x de l'article en cours de traitement

### **9.7.3 LNK.SRB, LNK(x).SRB**

Contient le nom de la RUBRIQUE du lien N° x de l'article en cours de traitement

### **9.7.4 LNK.SSRB, LNK(x).SSRB**

Contient le nom de la SOUS- du lien N° x de l'article en cours de traitement

### **9.7.5 LNK.ARTINTNB, LNK(x).ART.INTNB**

Contient le NUMERO de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.6 LNK.SRT, LNK(x).SRT**

Contient le SUR-TITRE de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.7 LNK.THM, LNK(x).THM**

Contient le THEME de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.8 LNK.TIT, LNK(x).TIT**

Contient le TITRE de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.9 LNK.LEADIN, LNK(x).LEADIN**

Contient le CHAPEAU de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.10 LNK.TXT, LNK(x).TXT**

Contient le TXT de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.7.11 LNK.SIG.XXX, LNK(x).SIG.XXX, LNK.SIG(y).XXX, LNK(x).SIG(y).XXX**

Préfixe permettant l'accès au champ XXX du signataire courant ou d'index y de l'article correspondant au lien N° x de l'article en cours de traitement.

## **9.8 Variables relatives aux encadrés**

On peut faire référence à l'encadré en cours (ce qui correspond au cas où l'on est en train de traiter une boucle sur les encadrés : cf. 8.9), ou bien à un encadré en particulier, en précisant son index dans la liste des encadrés associés à l'article en cours de traitement..

De manière générale, toute caractéristique de l'encadré pointé par le lien sont accessibles via le préfixe FRAME - (encadré courant) - ou FRAME(X) - (lien N°x) - : de la manière suivante :

### **9.8.1 FRAME.NBR**

Contient le nombre d'encadrés associés à l'article en cours.

### **9.8.2 FRAME.RUB, FRAME(x).RUB**

Contient le nom de la SECTION de l'encadré N° x de l'article en cours de traitement

### **9.8.3 FRAME.SRB, FRAME(x).SRB**

Contient le nom de la RUBRIQUE de l'encadré N° x de l'article en cours de traitement

### **9.8.4 FRAME.SSRB, FRAME(x).SSRB**

Contient le nom de la SOUS- de l'encadré N° x de l'article en cours de traitement

### **9.8.5 FRAME .ART.INTNB, FRAME(x).ART.INTNB**

Contient le NUMERO de l'encadré N° x de l'article en cours de traitement.

### **9.8.6 FRAME.SRT, FRAME(x).SRT**

Contient le SUR-TITRE de l'encadré N° x de l'article en cours de traitement.

### **9.8.7 FRAME.THM, FRAME(x).THM**

Contient le THEME l'encadré N° x de l'article en cours de traitement.

### **9.8.8 FRAME.TIT, FRAME(x).TIT**

Contient le TITRE de l'article correspondant au lien N° x de l'article en cours de traitement.

### **9.8.9 FRAME.LEADIN, FRAME(x).LEADIN**

Contient le CHAPEAU de l'encadré N° x de l'article en cours de traitement.

### **9.8.10 FRAME.TXT, FRAME(x).TXT**

Contient le TXT de l'encadré N° x de l'article en cours de traitement.

### **9.8.11 FRAME.SIG.XXX, FRAME(x).SIG.XXX, FRAME(x).SIG(y).XXX**

Préfixe permettant l'accès au champ XXX du signatire courant ou d'index y de l'encadré correspondant au lien N° x de l'article en cours de traitement.

## **9.9 Variables en Dur spéciales conditionnelles**

### **9.9.1 IF.SIG(x)**

Fait appel à la définition à appliquer si il existe une signature N°x associé à l'article en cours de traitement

### **9.9.2 IF.IMG(x)**

Fait appel à la définition à appliquer si il existe une image N°x associée à l'article en cours de traitement

### **9.9.3 IF.FILE(x)**

Fait appel à la définition à appliquer si il existe un fichier N°x est associé à l'article en cours de traitement

### **9.9.4 IF.LNK(x)**

Fait appel à la définition à appliquer si il existe un lien N°x associé à l'article en cours de traitement

### **9.9.5 IF.FRAME(x)**

Fait appel à la définition à appliquer si il existe un encadré N°x associé à l'article en cours de traitement

## 9.10 Boucle de sommaire : variable de « de Pattern »

### 9.10.1 Intérêt

Il est possible, dans une feuille de style, de spécifier des structures répétitives. On veut par exemple, dans la feuille de style d'un article, écrire la liste de toutes les rubriques de l'édition à laquelle il appartient. Il va donc falloir mettre en place une structure de boucle qui va s'appliquer à toutes les rubriques présentes dans l'édition.

### 9.10.2 Syntaxe

Une variable de pattern peut prendre les formes suivantes :

1. Simple : xxxPAT ou xxxLOOP
2. Développée : xxxPAT(expression) ou xxxLOOP(expression)
3. Développée avec suffixe : xxxPAT(expression).suffixe

Où xxx désigne le type d'objet sur lequel porte le pattern à savoir :

- RUBPAT (RUBLOOP) : pattern de Sections
- SRBPAT (SRBLOOP) : pattern de Rubriques
- SSRBPAT (SSRBLOOP) : pattern de Sous-Rubriques
- THMPAT (THMLOOP) : pattern de Thèmes
- KWDPAT (KWDLOOP) : pattern de mot-clé
- DATEARTPAT (DATEARTLOOP) : pattern de date d'article
- ARTPAT (ARTLOOP) : pattern d'articles
- SIGPAT (SIGLOOP) : pattern de signatures
- IMGPAT (IMGLOOP) : pattern d'images
- FILEPAT (FILELOOP) : pattern de fichier
- SELECTEDFILEPAT (SELECTEDFILELOOP) : pattern de fichiers ayant une extension spécifique `ext` qui doit être passé en argument.
- EXCLUDEDFILEPAT (SELECTEDFILELOOP) : pattern de fichiers ayant une extension différente d'une extension spécifique `ext` qui doit être passé en argument.
- FRAMEPAT (FRAMELOOP) : pattern d'Encadrés de l'article en cours
- ALLFRAMEPAT : pattern sur tous les encadrés
- LNKPAT (LNKLOOP) : pattern de Liens
- LINEPAT (LINELOOP) : pattern de ligne (pattern applicable à toutes les lignes du texte passé en paramètre.

### 9.10.3 Exemple pour une rubrique

**Exemple** : l'appel, dans la feuille de style d'un article appartenant à la section « TOTO » à la variable `!@RUBPAT(simple)@!` dont la définition de pattern de Rubrique est le suivant :

```
<!@RUBPAT(simple)>
  <!@MYSELF> Ma rubrique est : !@RUB@!
  <!/@MYSELF>
  <!@DEFAULT> je n'appartiens pas à la rubrique : !@RUB@!
  <!/@DEFAULT>
<!/@RUBPAT(simple)>
```

aura pour effet de remplacer cette variable par :

```
Je n'appartiens pas à la rubrique : TITI
```

```
Je n'appartiens pas à la rubrique : TUTU
Ma rubrique est la rubrique : TOTO
Je n'appartiens pas à la rubrique : TATA
```

#### 9.10.4 Exemple pour un article

*Exemple* : l'appel, dans la feuille de style de l'article appartenant à la sous-rubrique « SOUSRUB » de la rubrique « RUB » de la section « TOTO » à la variable !@ARTPAT(simple)@! dont la définition de pattern article est la suivante :

```
<!--@ARTPAT(simple)-->
<!--@ME-->
    JE SUIS L'ARTICLE COURANT : !@LOCNBA@!
<!--/@ME-->
<!--@DEFAULT-->
    je suis l'article N°: !@LOCNBA@!
<!--/@DEFAULT-->
<!--/@RUBPAT(simple)-->
```

aura pour effet, appliquer à l'article N°3 de cette sous-rubrique q(ui en contiendrait 5) , de remplacer l'appel au pattern par :

```
je suis l'article N°: 1
je suis l'article N°: 2
JE SUIS L'ARTICLE COURANT
je suis l'article N°: 4
je suis l'article N°: 5
```

#### 9.10.5 Imbrication de pattern

*Remarque* : le corps de texte des sous-balises des définitions de patterns peuvent contenir n'importe quelle autre variable RexPublisher, et entre autre des variable de pattern. Ainsi, si l'on veut pouvoir écrire le sommaire détaillé d'une édition il faudra emboîter les appels aux variables :

- RUBLOOP : dont le corps de définition contiendra à un appel à la variable
  - SRBLOOP : dont le corps de définition contiendra à un appel à la variable
    - SSRBLOOP : dont le corps de définition contiendra à un appel à la variable
      - ARTLOOP : dont le corps de définition contiendra à un appel à la variable

*Note* : Dans chaque « corps de boucle » le contexte « SECTION.RUBRIQUE.SOUS-RUBRIQUE » se trouve modifié. Ainsi, dans la mise en œuvre d'une boucle sur les sections (RUBLOOP), **la rubrique courante se trouve modifiée à chaque itération**. Ainsi, si son corps de pattern contient un appel à une boucle de Sous-rubrique (SRBLOOP), celle-ci ne s'effectuera que sur les sous-rubrique de la rubrique courante (i.e. la rubrique en cours de traitement dans le corps de RUBPLOOP).

### 9.10.6 Variables associées : les compteurs

Des variables (compteurs) sont associées à chaque contexte de boucle :

1. RUBLOOP.IDX : désigne le N° d'itération de la boucle sur les sections courantes
2. SRBLOOP.IDX : désigne le N° d'itération de la boucle sur les rubriques courantes
3. SSRBLOOP.IDX : désigne le N° d'itération de la boucle sur les sous-rubriques courantes
4. ARTLOOP.IDX : désigne le N° d'itération de la boucle sur les articles courants
5. DATEARTLOOP.IDX : désigne le N° d'itération de la boucle sur les articles courants
6. KWDLOOP.IDX : désigne le N° d'itération de la boucle sur les mots-clés courants
7. SIGLOOP.IDX : désigne le N° d'itération de la boucle sur les signatures courantes
8. IMGLOOP.IDX : désigne le N° d'itération de la boucle sur les images courantes
9. FILELOOP.IDX : désigne le N° d'itération de la boucle sur les fichiers courants
10. LNKLOOP.IDX : désigne le N° d'itération de la boucle sur les liens courants
11. FRAMELOOP.IDX : désigne le N° d'itération de la boucle sur les encadrés courants
12. THMLOOP.IDX : désigne le N° d'itération de la boucle sur les thèmes courants

## 9.11 Fonctions RexPublisher

### 9.11.1 ALPHAWIDTH

Format : ALPHAWIDTH( [ SecIdx[ , RubIdx ] ] )

Effet : renvoie :

- la longueur du nom de la sous-rubrique la plus longue de la rubrique d'index **RubIdx** de la section d'index **SecIdx**. Si ces 2 paramètres sont précisés
- la longueur du nom de la rubrique la plus longue de la section d'index **SecIdx**. Si ce paramètre est précisé
- la longueur du nom de la section la plus longue si aucun paramètre n'est précisé

### 9.11.2 AND

Format : AND( cond1 , cond2 )

Effet : ET logique sur **cond1** et **cond2**

### 9.11.3 ART.INDEX

Format : ARTPOS( SecIdx[ , RubIdx[ , SrubIdx ] ] )

Effet : renvoie la position de l'article courant dans la la sous-rubrique d'index **SrubIdx** de la rubrique d'index **RubIdx** de la section d'index **SecIdx**.

### 9.11.4 ARTNBR

Format : ARTNBR( I , J , K )

Effet : renvoie le nombre d'article contenus dans la sous-rubriques **K** de la rubrique **J** de la section **I**.

**ou**

Format :ARTNBR( I , J )

Effet : renvoie le nombre d'article contenus dans la rubrique **J** de la section **I**.

**ou**

Format : ARTNBR( I )

Effet : renvoie le nombre d'article contenus dans la section d'index **I**.

### 9.11.5 ART.POS

Format : ARTPOS( P )

En fonction de la valeur de l'argument **P** entre paranethèse, renvoie Contient le NUMERO correspondant à la position de l'article en cours dans :

- P = 0 : dans l'édition
- P = 1 : dans sa section
- P = 2 : dans sa rubrique
- P = 3 ou vide : dans sa sous-rubrique

### 9.11.6 ASCII

Format : ASCII( char )

Effet : renvoie le code ASCII du **caractère** passé en paramètre

New !

**New !**

### 9.11.7 ASKUSER

Format : ASKUSER(*typedialogue*, *titre*, *valeurs*, *valeurpardéfaut*)

Effet : permet d'introduire un dialogue avec l'utilisateur, au travers duquel celui-ci va pouvoir choisir des valeurs. Les paramètres en sont les suivants :

- **Typedialogue** : il permet de définir le type de dialogue :
  - *Text* : permet à l'utilisateur de saisir un texte
  - *List* : permet de sélectionner des valeurs multiples dans une liste
  - *Combo* : permet de sélectionner une valeur dans un menu déroulant
  - *Bool* : permet de sélectionner une valeur entre deux (bouton radio)
- **Titre** : permet de définir le titre de la fenêtre de dialogue
- **Valeurs** : permet de définir les valeurs proposées à l'utilisateur, les valeurs sont séparées par le caractère « | » (pipe) :
  - *Text* : non applicable
  - *List* : ex : a|b|c
  - *Combo* : idem
  - *Bool* : ex : rouge|Noir
- **valeurpardéfaut** : elle permet de définir, parmi la liste de valeur définies précédemment, la ou les valeurs proposées par défaut.

### 9.11.8 BREAK.ON

Format : BREAK.ON

Effet : Autorise le traitement des sauts de lignes (Cf. BRKSTY).

### 9.11.9 BREAK.OFF

Format : BREAK.ON

Effet : masque le traitement des sauts de lignes (Cf. BRKSTY).

**New !**

### 9.11.10 CREATEPDF

Format : CREATEPDF(*ORIGIN*, *DESTINATION*)

Effet : crée le fichier PDF *destination* à partir du fichier *origin*.

Note : cette fonction nécessite que WIN2PDF (www.win2pdf.com) ou Acrobat PDFWriter soient installés sur le poste client.

**New !**

### 9.11.11 CHR

Format : CHR(*ASCII*)

Effet : renvoie le caractère correspondant au code Ascii fourni en paramètre.

### 9.11.12 CURR\_RUBNBR

Format : CURR\_RUBNBR

Effet : renvoie le nombre de sections connues de l'édition en cours contenant des articles.



9.11.13 CURR\_SRBNBR

Format : CURR\_SRBNBR

Effet : renvoie le nombre de rubriques connues de la section courante de l'édition en cours contenant des articles.

ou

Format : CURR\_SRBNBR ( I )

Effet : renvoie le nombre de rubriques connues de la section d'index **I** de l'édition en cours contenant des articles.

9.11.14 CURR\_SSRBNBR

Format : SSRBNBR

Effet : renvoie le nombre de sous-rubriques connues de la rubrique courante de la section courante de l'édition en cours (qu'elles contiennent ou non des articles).

ou

Format : CURR\_SSRBNBR(I,J)

Effet : renvoie le nombre de sous-rubriques connues de la rubrique d'index **J** de la section d'index **I** l'édition en cours (qu'elles contiennent ou non des articles).

New !

9.11.15 CREATEPDF

Format : CREATEPDF ( filename , PDFfilename )

Effet : Crée un PDF nommé **PDFfilename** à partir du fichier **Filename**.

**ATTENTION** : cette fonctionnalité n'est disponible que si l'utilisateur dispose de l'imprimante Win2PDF ou Adobe PDFWriter sur son poste.

9.11.16 DATEADD

Format : DATE ( interval , number , date )

Effet : retourne une date calculée à partir des arguments suivant

- **Interval** : est l'intervalle de temps que l'on veut ajouter :

Les valeurs possibles pour **interval** sont :

yyyy	année
q	trimestre
m	mois
y	jour
d	jour
w	jour
ww	semaine
h	heure
n	Minute
s	Seconde

- **number** : nombre d'intervalles que l'on veut rajouter
- **date** : date à laquelle le nombre d'intervalle de temps est rajouté

### 9.11.17 DAY

Format : DAY(date)

Effet : retourne le jour date de la date [date](#) .

### 9.11.18 DAYNAME

Format : DAYNAME(date)

Effet : retourne le nom du jour date de la date [date](#) .

### 9.11.19 DIFFERENT

Format : DIFFERENT(exp1,exp2)

Effet : renvoie « [TRUE](#) » si les expressions passées en paramètres sont différentes et « [FALSE](#) » sinon.

### 9.11.20 DIVIDE

Format : DIVIDE(exp1,exp2)

Effet : renvoie la division (arithmétique) de [exp1](#) par [exp2](#).

### 9.11.21 DOC\_INSERTTOC

Format : DOC\_INSERTTOC (Fichier,Signet)

Effet : insert une table de matière dans le document word [Fichier](#) à l'emplacement du signet [Signet](#).

### 9.11.22 EQUAL

Format : EQUAL(exp1,exp2)

Effet : renvoie « [TRUE](#) » si les expressions passées en paramètres sont égales et « [FALSE](#) » sinon.

### 9.11.23 EVEN

Format : EVEN(exp1)

Effet : renvoie « [TRUE](#) » si [Exp1](#) est paire et « [FALSE](#) » sinon.

### 9.11.24 EXISTS

Format : EXISTS( Sec [ , Rub [ , Srub ] ] )

Effet : suivant la nombre d'arguments, renvoie « [TRUE](#) » si la sous-rubrique [Srub](#) de la rubrique [Rub](#), de la Section [Sec](#) contient un article

### 9.11.25 FILE\_EXTENTION

Format : FILE\_EXTENTION(filename)

Effet : renvoie l'extension du fichier passé en paramètre

### 9.11.26 FILESIZE

Format : FILESIZE(filename)

Effet : renvoie la taille en KO du fichier passé en paramètre

New !

New !

9.11.27 FILEMODIFIED

New !

Format : FILEMODIFIED( filename )

Effet : renvoie la date de modification du fichier passé en paramètre

9.11.28 FILEPATH

Format : FILEPATH

Effet : renvoie le chemin d'accès aux fichiers associés à l'article courant.

New !

9.11.29 FLOOR

Format : FLOOR( Num )

Effet : renvoie la partie entière du nombre passé en paramètre.

New !

9.11.30 FORMAT

Format : FORMAT( String, FormatStr )

Effet : permet de formater la chaîne **String** (qui peut être un nombre ou une date) suivant le format défini dans **FormatStr**.

Les symboles utilisables dans la chaîne de formatage sont les suivants :

Symbole	Description
0	Imprime un chiffre ou un zéro dans cette position si nécessaire
#	Imprime un chiffre dans cette position si nécessaire
.	Position, de la virgule
,	Position de la séparation des milliers
- + \$ ( ) espace	Caractère affiché exactement dans la position spécifiée.

Exemples :

syntaxe	Résultat
Format(8315.4, "00000.00")	08315.40
Format(8315.4, "#####.##")	8315.4
Format(8315.4, "##,##0.00")	8,315.40
Format(315.4, "\$##0.00")	\$315.40

9.11.31 GETSLICE

Format : GET\_SLICE( SliceIdx, SliceNb, ExpString[ , Critéria ] )

Effet : cette fonction :

- Coupe **ExpStrin** en autant de chaînes séparées par la chaîne **Criteria**.
- Regroupe toutes ces tranches en **SliceNb** paquets (le dernier étant éventuellement plus petit

- Renvoie le paquet (chaîne) [SliceIdx](#) (si celui-ci est plus grand que 0 et plus petit ou égal à [SliceNb](#)).

**New !**

### 9.11.32 GENPATH

Format : GENPATH

Effet : renvoie le chemin d'accès aux répertoire de génération courant.

**New !**

### 9.11.33 GETVAR

Format : GETVAR ( VarName )

Effet : renvoie la valeur stockée dans la variable nommée [Varname](#) (voir SETVAR).

### 9.11.34 IMGPATH

Format : IMGPATH

Effet : renvoie le chemin d'accès aux images associées à l'article courant.

### 9.11.35 INSERTEDIMGPATH

Format : INSERTEDIMGPATH

Effet : renvoie le chemin d'accès aux images incluses dans le texte de l'article courant.

**New !**

### 9.11.36 INSERTPDF

Format : INSERTPDF ( Pdf1 , Pdf2 , Pdf3 )

Effet : insère le document PDF décrit par le chemin [Pdf2](#), à la fin du document PDF décrit par le chemin [Pdf1](#), et le sauve sous la forme d'un document PDF décrit par le chemin [Pdf3](#).

Note : [Pdf1](#) et [Pdf3](#) peuvent décrire le même chemin.

### 9.11.37 LOCATION

Format : LOCATION

Effet : renvoie le chemin complet du fichier généré.

**New !**

### 9.11.38 LENGTH

Format : LENGTH ( exp1 )

Effet : renvoie la longueur de la chaîne [Exp1](#).

### 9.11.39 LESS

Format : LESS ( exp1 , exp2 )

Effet : renvoie « [TRUE](#) » si [Exp1](#) < [Exp2](#) et « [FALSE](#) » sinon.

**New !**

### 9.11.40 LIKE

Format : LIKE ( pattern , chaine )

Effet : Compare les 2 chaînes passées en paramètres, suivant des règles dites de « pattern matching ». Il renvoie « [TRUE](#) » si [chaine](#) « [matche](#) » [Pattern](#) et « [FALSE](#) » sinon.

Le pattern matching fournit un outil pour les comparaisons de chaînes. Les critères spéciaux permettent de trouver des correspondances aux chaînes à l'aide de caractères génériques,

de listes de caractères ou de plages de caractères, dans n'importe quelle combinaison. Le tableau suivant montre les caractères autorisés dans pattern et ce à quoi ils correspondent :

Caractères dans pattern	Correspond dans string à
?	Tout caractère unique
*	Zéro ou plusieurs caractères
#	Tout chiffre (0-9)
[charlist]	Tout caractère présent dans charlist
[!charlist]	Tout caractère non présent dans charlist

Un groupe d'un ou plusieurs caractères (charlist) placés entre crochets ([ ]) peut être utilisé pour faire correspondre tout caractère unique dans string et peut comprendre presque tous les codes de caractères, y compris les chiffres.

*Remarque* : Pour trouver des correspondances aux caractères spéciaux crochet gauche ([), point d'interrogation (?), signe dièse (#) et astérisque (\*), placez-les entre crochets. Le crochet droit (]) ne peut pas être utilisé dans un groupe pour correspondre à lui-même, mais il peut l'être à l'extérieur d'un groupe en tant que caractère individuel.

En utilisant un trait d'union (–) pour séparer les limites inférieure et supérieure de la plage, charlist peut spécifier une plage de caractères. Par exemple, [A-Z] génère une correspondance si la position des caractères correspondante dans string contient toutes les lettres majuscules de la plage A–Z. Plusieurs plages sont comprises dans les crochets sans délimiteurs.

Exemples :

- LIKE(F,F) renvoie TRUE.
- LIKE(aBBBa,a\*a) renvoie TRUE.
- LIKE(F,[A-Z]) renvoie TRUE.
- LIKE(F,[!A-Z]) renvoie FALSE.
- LIKE(a2a,a#a) renvoie TRUE
- LIKE(aM5b,a[L-P]#[!c-e]) renvoie FALSE

#### 9.11.41 LOOP

Format : LOOP(debut, fin[,pas])

Effet : Loop est une fonction de boucle du type :

Do from **debut** to **Fin**, setp **pas**

où :

- **debut** est une expression à évaluer donnant la valeur initiale de la variable de boucle,
- **fin** est une expression à évaluer donnant la valeur finale de la variable de boucle,
- **pas** est une expression à évaluer donnant le pas à appliquer dans l'incrément de la variable de boucle (par défaut 1).

L'effet de LOOP sera d'écrire autant de fois que la structure boucle le spécifiera la valeur associée, dans la feuille de style à la définition de LOOP (il est donc conseillé d'associer un suffixe à la définition.

On peut faire référence, dans le corps de définition, à la valeur courante de la variable d'itération à l'aide de la variable générique : !@@!

Exemple :

Pour l'appel à : !@LOOP(3,5).maboucle@ !

on utilise la définition

```
<!@LOOP(?).maboucle>
<!@DEFAULT>
    la valeur courante est : !@@! <br>
<!/@DEFAULT>
<!/@LOOP(?).maboucle>
```

ceci aura pour effet d'écrire dans le document généré

```
la valeur courante est : 3
la valeur courante est : 4
la valeur courante est : 5
```

#### 9.11.42 LOWER

Format : LOWER(expression1)

Effet : renvoi [expression1](#) en bas de casse.

#### 9.11.43 MINUS

Format : MINUS(exp1,exp2)

Effet : renvoie la différence (arithmétique) de [exp1](#) et [exp2](#).

#### 9.11.44 MODULO

Format : MODULO(exp1,exp2)

Effet : renvoie la valeur de [exp1](#) Modulo [exp2](#).

#### 9.11.45 MONTH

Format : MONTH(date)

Effet : retourne le mois de la date [date](#) .

#### 9.11.46 MONTHNAME

Format : MONTHNAME(date)

Effet : retourne le nom du mois de la date [date](#) .

#### 9.11.47 MORE

Format : MORE(exp1,exp2)

Effet : renvoie « TRUE » si [Exp1](#) >= [Exp2](#) et « FALSE » sinon.

New !

#### 9.11.48 MOVEFILE

Format : MOVEFILE(Filepath1 , Filepath2)

Effet : copie le fichier [Filepath1](#) en, [Filepath2](#) .

#### 9.11.49 MULTIPLY

Format : MULTIPLY(exp1,exp2)

Effet : renvoie la multiplication (arithmétique) de [exp1](#) par [exp2](#).

#### 9.11.50 NOBLANK

Format : NOBLANK([expression1](#))

Effet : renvoi [expression1](#) débarrassé de tous ses espaces blancs

New !

#### 9.11.51 NODEINDEX

Format : NODEINDEX( Sec [ , Rub [ , Srub ] ] )

Effet : suivant la nombre d'arguments, renvoie l'index du nœud de l'arborescence caractérisé par ses paramètre (sousbrique [Srub](#) de la rubrique [Rub](#), de la Section [Sec](#)) par rapport au nœud qui le contient :

Exemples :

- NODEINDEX(toto) renvoie l'index de la section Toto dans l'édition courante
- NODEINDEX(toto,Titi) renvoie l'index de la rubrique Titi dans la section Toto dans l'édition courante

New !

#### 9.11.52 NODENAME

Format : NODEINDEX( SecIdx [ , RubIdx [ , SrubIdx ] ] )

Effet : suivant la nombre d'arguments, renvoie le nom du nœud de l'arborescence caractérisé par ses paramètre (sousbrique d'index [SrubIdx](#) de la rubrique d'index [RubIdx](#), de la Section d'index [SecIdx](#)) par rapport au nœud qui le contient :

Exemples :

- NODENAME(1) renvoie l'index de la 1<sup>ère</sup> section dans l'édition courante
- NODENAME (1,2) renvoie l'index de la rubrique d'index 2 dans la section d'index 1 dans l'édition courante

ATTENTION : les Rubriques ou sous Rubrique de nom « Vide » ont pour index 0.

#### 9.11.53 ODD

Format : ODD([exp1](#),[exp2](#))

Effet : renvoie « TRUE » si [Exp1](#) est impaire et « FALSE » sinon.

#### 9.11.54 OR

Format : ODD([exp1](#),[exp2](#), [[exp3](#),...])

Effet : renvoie « TRUE » si un des arguments est vrai « FALSE » sinon.

#### 9.11.55 PLUS

Format : PLUS([exp1](#),[exp2](#))

Effet : renvoie la somme (arithmétique) de [exp1](#) et [exp2](#).

New !

#### 9.11.56 PS\_ACTIONS

Format : PS\_ACTIONS([Imgpath](#), Action1, Action2, ....)

Effet : permet d'exécuter sur le fichier [Imgpath](#) les actions photoshop listées à sa suite..

Note : cette fonctionnalité nécessite la présence de Photoshop sur la machine de l'utilisateur.

### 9.11.57 PROCESSXLS

Format : PROCESSXLS(exp1)

Effet : lance la transformation du fichier excel [exp1](#) (ne fait rien si exp1 n'est pas un fichier excel) Voir 1.31

New !

### 9.11.58 PUBPATH

Format : PUBPATH

Effet : renvoie le chemin d'accès à la publication courante.

New !

### 9.11.59 READFILE

Format : READFILE(FileName)

Effet : renvoie le contenu du fichier [Filename](#).

New !

### 9.11.60 READPASTESAVE

Format : READPASTESAVE(filename1,Filename2,Filename3 [,Position])

Effet : cette fonction permet de copier le fichier [Filename2](#) dans le Fichier template [Filename1](#) à la position (optionnelle) [Position](#), pour le sauver sous le nom [Filename3](#).

Position peut prendre les valeurs [End](#) (fin de fichier), ou [Begin](#) (Début de fichier). La valeur par défaut est [End](#).

*Note* : cette fonction est particulièrement utile pour créer un fichier Word à partir d'un fichier HTML. La fonction nécessite la présence d'au minimum Word2000 sur la machine de l'utilisateur.

### 9.11.61 RELATIVEPATH

Format : RELATIVEPATH(filename1,Filename2)

Effet : renvoie le chemin relatif de [filename1](#) par rapport au chemin de [filename2](#)

New !

### 9.11.62 REMOVEFILE

Format : REMOVEFILE(Filepath1)

Effet : supprime le fichier [Filepath1](#).

### 9.11.63 REPLACE

Format : REPLACE(chaine, chercher, remplacer)

Effet : retourne la chaîne de caractère [chaine](#), où l'on a remplacé la chaîne [chercher](#) par la chaîne [remplacer](#).

### 9.11.64 REPLACEBLANK

Format : REPLACEBLANK(chaine, remplacer)

Effet : retourne la chaîne de caractère [chaine](#), où l'on a remplacé tous les espaces blancs par la chaîne [remplacer](#).

New !

### 9.11.65 RESIZE

Format : RESIZE(ImgPath, Width, Height)

Effet : permet de redimensionner l'image. [ImgPath](#) à la taille définie par [Width](#) et [Height](#).



Note : si **Width** ou **Height** sont égaux à « \_ » (blanc souligné), il suffit que l'autre paramètre soit défini pour qu'un redimensionnement proportionnel soit effectué.

**New !**

#### 9.11.66 ROUND

Format : ROUND( exp1 , exp2 )

Effet : permet d'arrondir l'expression numérique **Exp1** à un certain nombre de chiffres après la virgule défini par **Exp2**.

#### 9.11.67 RUBNBR

Format : RUBNBR

Effet : renvoie le nombre de sections connues de l'édition en cours (qu'elles contiennent ou non des articles).

**New !**

#### 9.11.68 SETVAR

Format : SETVAR( VarName,Value)

Effet : stocke la valeur Value dans la variable nommée **Varname** (voir GETVAR).

Note : cette fonction est particulièrement utile pour stocker une valeur calculée pendant l'interprétation d'une feuille de style. Cette valeur peut être récupérée à l'aide la fonction GETVAR.

#### 9.11.69 SRBNBR

Format : SRBNBR

Effet : renvoie le nombre de rubriques connues de la section courante de l'édition en cours (qu'elles contiennent ou non des articles).

**ou**

Format : SRBNBR ( I )

Effet : renvoie le nombre de rubriques connues de la section d'index **I** de l'édition en cours (qu'elles contiennent ou non des articles).

#### 9.11.70 SSRBNBR

Format : SSRBNBR

Effet : renvoie le nombre de sous-rubriques connues de la rubrique courante de la section courante de l'édition en cours (qu'elles contiennent ou non des articles).

**ou**

Format : SSRBNBR ( I , J )

Effet : renvoie le nombre de sous-rubriques connues de la rubrique d'index **J** de la section d'index **I** de l'édition en cours (qu'elles contiennent ou non des articles).

#### 9.11.71 SUBSTR

Format : SUBSTR( chaîne, I, L)

Effet : retourne la sous-chaîne de caractère chaîne, construite à partir du caractère N° **I** de la chaîne chaîne sur une longueur **L**.

### 9.11.72 THIS

Format : THIS (Exp1)

Effet : retourne l'évaluation de Exp1.

### 9.11.73 TODAY!

Format : TODAY

Effet : retourne la date du jour

### 9.11.74 TRANSCODE

Format : TRANSCODE (Code, Expression)

Effet : renvoie la chaîne Expression transcodé suivant le code Code .

Les codes possibles sont :

- Html, js, noaccent, url

### 9.11.75 TRANSCODE.OFF

Format : TRANSCODE.OFF

Effet : annule le transcodage automatique effectué par RexPublisher des caractères spéciaux, dont l'effet est fonction de l'extension du fichier généré. .

### 9.11.76 TRANSCODE.ON

Format : TRANSCODE.ON

Effet : rétablit le transcodage automatique effectué par RexPublisher des caractères spéciaux, dont l'effet est fonction de l'extension du fichier généré. .

### 9.11.77 TRIM

Format : TRIM (exp1)

Effet : renvoie la chaîne tronquée des caractères blancs du début et de la fin.

New !

### 9.11.78 UNWRAP

Format : UNWRAP (expression1)

Effet : renvoie expression1 dénuée de tout retour à la ligne.

### 9.11.79 UPPER

Format : UPPER (expression1)

Effet : renvoie expression1 en haut de casse.

### 9.11.80 URLENCODE

Format : URLENCODE (expression)

Effet : renvoie l'URL-encodage d'expression.

### 9.11.81 VALIDFILENAME

Format : VALIDFILENAME (expression1)

Effet : renvoi une évaluation de `expression1` où tous les caractères non autorisés dans un nom de fichier ont été remplacé par un underscore (blanc souligné).

**New !**

#### 9.11.82 VALIDFILEPATH

Format : `VALIDFILEPATH(expression1)`

Effet : renvoi une évaluation de `expression1` où tous les caractères non autorisés dans un chemin de fichier ont été remplacé par un underscore (blanc souligné).

**New !**

#### 9.11.83 WRITE\_IF

Format : `WRITE_IF(Bool , Exp1 [, Exp2] )`

Effet : renvoie Exp1 si `Bool` est égal à `True`, sinon renvoie Exp2 si ce paramètre est présent.

**New !**

#### 9.11.84 WRITE\_FILE

Format : `WRITE_FILE(fileName, chaîne)`

Effet : écrit la chaîne `Chaîne` dans le fichier `Filename`.

#### 9.11.85 YEAR

Format : `YEAR(date)`

Effet : retourne l'année de la date `date`.

## 9.12 Fonction d'Accès à la base de données de la Publication

Il est possible, dans une feuille de style, de faire référence aux éditions précédentes de la publication. Ceci s'effectue à l'aide des fonctions dédiées suivantes.

### 9.12.1 Fonctions d'accès

#### 9.12.1.1 DBART

**Format** : DBARTLOOP[ (X) [ .suffixe ] ]

Où X est un nombre entier

**Effet** : effectue une opération du type PATTERN :

- sur les X derniers articles si X est un nombre entier
- sur tous les articles sinon (y compris si X est absent)

archivés dans la base dont la Section, Rubrique, Sous-Rubrique correspondent aux sections, rubriques, et sous-rubriques courantes résultant de l'interprétation de la feuille de style au moment de l'appel.

**Exemple** : imaginons la balise de pattern de Section suivante :

```
< !@RUBLOOP (BD) >
< !@TOTO>
  !@DBARTLOOP(5) .essai@!
< !/@TOTO>
< !/@RUBLOOP (BD) >
```

l'appel dans la feuille de style à la variable !@RUBLOOP(BD)@! Aura pour effet de ne s'appliquer que sur section TOTO, puisque aucune définition n'est prévu pour les autres.

Le corps de cette sous-balise contenant un appel à la fonction DBARTLOOP reviendra à aller chercher dans la base de donnée les 5 derniers articles dont la rubrique est TOTO.

Ces 5 derniers articles correspondent très exactement à un *recordset* tel qu'il pourrait être renvoyé par une requête SQL sur cette même base.

#### 9.12.1.2 DBART

**Format** : DBARTPAT(X) [ .suffixe ] ]

Où X est un nombre entier

**Effet** : effectue une opération du type PATTERN sur le Xième dernier article archivé dans la base dont la Section, Rubrique, Sous-Rubrique correspondent aux sections, rubriques, et sous-rubriques courantes résultant de l'interprétation de la feuille de style au moment de l'appel.

#### 9.12.1.3 DBSQL

**Format** : DBSQL(requete) [ .suffixe ]

Où l'évaluation de *requete* est un requête SQL valide

**Effet** : effectue une opération du type PATTERN sur les articles archivés dans la base dont correspondant aux critères exprimés dans la requête SQL..

### 9.12.2 Variables d'accès

L'accès à la base de données s'effectuant au travers d'un mécanisme de Pattern, il est nécessaire de référencer explicitement, dans lesdits pattern, les champs des articles retournés par les fonctions précédentes. Ceci s'effectue au travers des variables spécialisées suivantes.

#### 9.12.2.1 *DB.EDITION.NB*

Contient le N° D'EDITION de l'article accédé dans la BD en cours de traitement

#### 9.12.2.2 *DB.DATE*

Contient la DATE de l'article accédé dans la BD en cours de traitement

#### 9.12.2.3 *DB.RUB*

Contient le nom de la SECTION de l'article accédé dans la BD en cours de traitement

#### 9.12.2.4 *DB.SRUB*

Contient le nom de la RUBRIQUE de l'article accédé dans la BD en cours de traitement

#### 9.12.2.5 *DB.SSRUB*

Contient le nom de la SOUS- de l'article accédé dans la BD en cours de traitement

#### 9.12.2.6 *DB.ART.INTNB*

Contient le NUMERO de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.7 *DB.SRT*

Contient le SUR-TITRE de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.8 *DB.THM*

Contient le THEME l'encadré N° x de l'article en cours de traitement.

#### 9.12.2.9 *DB.TIT*

Contient le TITRE de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.10 *DB.LEADIN*

Contient le CHAPEAU de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.11 *DB.TXT*

Contient le TXT de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.12 *DB.SIG*

Contient la signature de l'article accédé dans la BD en cours de traitement.

#### 9.12.2.13 *DB.IMG(x).NAME*

Contient le nom de l'image N°X associé à l'article accédé dans la BD en cours de traitement.

#### 9.12.2.14 *DB.IMG(x)CAPTION*

Contient la légende de l'image N°X associé à l'article en cours accédé dans la BD.

#### 9.12.2.15 *DB.IMG(x)COPYRIGHT*

Contient le copyright de l'image N°X associé à l'article en cours accédé dans la BD.

#### 9.12.2.16 *DB.IMG(x).LINK*

Contient le lien de l'image N°X associé à l'article en cours accédé dans la BD.

### 9.12.2.17 Variables conditionnelles

Chacune des variables XXX précédentes à son pendant conditionnel sous la forme :

!@IF.DB.XXX [ (expression)[.suffixe] ]

ou :

!@DB.XXXSTY [ (expression)[.suffixe] ]

### 9.13 Accès à d'autres bases de données

Il est aussi possible, dans une feuille de style, de faire à des données extraites d'autres bases de données ACCESS que celle de la publication. Ceci s'effectue à l'aide des fonctions dédiées suivantes.

#### 9.13.1 EXOSQL

Format : EXOSQL(BD, (requete)) [ .suffixe ]

Où :

- BD décrit le chemin d'une base de donnée ACCESS.
- l'évaluation de `requete` est un requête SQL valide

Effet : effectue une opération du type PATTERN sur les articles archivés dans la base BD correspondant aux critères exprimés dans la requête SQL..

#### 9.13.2 Variables d'accès aux BD externes

L'accès à cette base de données s'effectuant au travers d'un mécanisme de Pattern, il est nécessaire de référencer explicitement, dans lesdits pattern, les champs des articles retournés par les fonctions précédentes.

Ce référencement se fait par le biais de variables spéciales dont la forme est :

!@DB.FIELDx@!

Où x est un indice variant de 1 à 30.

Il est alors nécessaire de créer des alias vers ces variables dans la requete SQL trransmise en argument à la fonction DBSQL :

Exemple : l'appel à la variable

```
!@EXOSQL("c:/mabd/toto.mdb",
  (SELECT Nom as Fiedl1, Prenom as Field2
   FROM Table1
   WHERE Field1 LIKE "P*") ).extro@ !
```

fera appel à la définition :

```
<!/EXOSQL(?).extro>
<!/DEFAULT>
  Nom : !@DB.FIELD1@!
  Prénom : !@DB.FIELD2@!
<!/DEFAULT>
<!/EXOSQL(?).extro
```

## 9.14 Accès au contenu d'une feuille Excel

On peut faire référence aux cellules de la feuille excel associée à un article dans la feuille style.

### 9.14.1 Fonction d'appel ; PROCESSXLS

On appelle l'exploitation d'une feuille EXCEL à l'aide de la fonction : PROCESSXLS

Format : PROCESSXLS (filename1)

Effet : lance l'interprétation de la feuille [filename1](#).

Cette exploitation est effectuée en recherchant et en interprétant les balise suivantes :

### 9.14.2 Balise <!/ @TABLE.BEGIN>

Cette balise permet de définir le *bloc de début* de la mise en forme de la table Excel

Exemple :

```
< !@TABLE . BEGIN>
< !@DEFAULT>
    <br>
    <table width="99%" border="1" cellspacing="0"
        cellpadding="2">
< !/@DEFAULT>
< !/@TABLE . BEGIN>
```

### 9.14.3 Balise <!/ @TABLE.END>

Cette balise permet de définir le *bloc de fin* de la mise en forme de la table Excel.

Exemple :

```
< !@TABLE . ROW . BEGIN>
< !@DEFAULT>
    </table>
< !/@DEFAULT>
< !/@TABLE . ROW . BEGIN>
```

### 9.14.4 Balise <!/ @TABLE.ROW.BEGIN >

Cette balise permet de définir le *bloc de début* de la mise en forme d'une ligne de la table.

Exemple :

```
< !@TABLE . ROW . BEGIN>
< !@DEFAULT>
    <TR class="text">
< !/@DEFAULT>
< !/@TABLE . ROW . BEGIN>
```



### 9.14.5 Balise `<!/ @TABLE.ROW.END>`

Cette balise permet de définir le *bloc de fin* de la mise en forme d'une ligne de la table.

Exemple :

```
< !@TABLE . ROW . END >
< !@DEFAULT >
  < /TR >
< ! /@DEFAULT >
< ! /@TABLE . ROW . END >
```

### 9.14.6 Balise `<!/ @TABLE.CELL(X,Y)>`

Cette balise permet de définir le traitement à appliquer à la cellule de coordonnée X,Y.

Note : X et Y peuvent avoir les valeurs suivantes !

- Numérique : elle désigne dans ce cas une coordonnée précise
- La valeur \* : c'est un « joker » classique, qui désigne toutes les valeurs possibles de la coordonnée.

Exemple :

```
< !@TABLE . CELL ( 1 , 1 ) >
< !@DEFAULT >
  < td class="texteblanc" >
    < div align="center" >< b> !@@! < /b >< /div >
  < /td >
< ! /@DEFAULT >
< ! /@TABLE . CELL ( 1 , 1 ) >

< !@TABLE . CELL ( * , 1 ) >
< !@DEFAULT >
  < td class="texteblanc" bgcolor="#003366" >
    < div align="center" >< b> !@@! < /b >< /div >
  < /td >
< ! /@DEFAULT >
< ! /@TABLE . CELL ( * , 1 ) >
```

### 9.14.7 Balise `<!/ @TABLE.VOID(x,y)>`

Cette balise permet de définir le traitement à appliquer à la cellule de coordonnée X,Y, **si elle est vide**.

Note : X et Y peuvent avoir les valeurs suivantes !

- Numérique : elle désigne dans ce cas une coordonnée précise
- La valeur \* : c'est un « joker » classique, qui désigne toutes les valeurs possibles de la coordonnée.

Exemples :

```
< !@TABLE . VOID ( * , * ) >
< !@DEFAULT >
  < td class="textebleu" >
    < div align="center" > !@@! &nbsp; < /b >< /div >< /td >
< ! /@DEFAULT >
< ! /@TABLE . VOID ( * , * ) >
```