

# **georchestra**

**version 1.0**

Camp tocamp

April 26, 2012



# Contents

<b>Fonctionnalités principales</b>	<b>1</b>
<b>Une réponse efficace à la directive INSPIRE</b>	<b>2</b>
<b>Implémentation des standards</b>	<b>2</b>
<b>Une communauté d'utilisateurs et de développeurs</b>	<b>2</b>
Documentation	2
Gouvernance	3
Comité de Pilotage du Projet (PSC)	3
Introduction	3
Qu'est-ce que geOrchestra ?	3
La philosophie de développement	3
Responsabilité du PSC	3
Observations	4
Membres du PSC	4
Ajout de membres au PSC	4
Retrait d'un membre du PSC	4
Responsabilités des membres du PSC	4
Conduite des développements	4
Participation aux réunions	4
Participation aux listes de diffusion	4
Gestion des RFC	5
Comment cela fonctionne	5
Vote d'une RFC	5
Implémentation d'une RFC	6
Quand le vote est-il obligatoire ?	6
Présentation	6
Aperçu des fonctionnalités	7
Architecture	7
L'approche par modules	7
Le respect des normes	7
Le suivi de la charge	7
Visualiseur	7
Styleur	8
Requêteur	9
Impression	10
Éditeur	10
Catalogue	11

Extracteur	12
Serveur Cartographique	13
Geopublisher	13
Fiche d'identité du projet geOrchestra	13
Portail public (CMS)	17
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	17
Date de première version	17
Nombre de versions parues	17
Nombre d'utilisateurs / installations	17
Quel pilotage des évolutions ?	17
Quel financement des évolutions ?	17
Le CMS comprend-il une gestion de groupes de travail ?	17
Quelle est la liaison avec le catalogue de métadonnées ? (affichage des dernières MD créées, des MD les plus consultées ...)	17
Y a-t-il localisation cartographique des contenus du CMS ?	17
Administration du portail (CMS)	17
Existe-t-il un module de gestion des conventions d'échange ?	17
Quelle est la granularité de gestion des droits ? (par contenus, par fonctionnalité, sur les fichiers...)	18
Comment est organisée la documentation (arborescence de fichiers ? Bases de données documentaire ?) et quelle liaison avec les éléments du CMS ?	18
Comment se fait la restitution des statistiques de consultation ?	18
Outil de visualisation cartographique	18
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	18
Date de première version	18
Nombre de versions parues	18
Nombre d'utilisateurs / installations	18
Quel pilotage des évolutions ?	18
Quel financement des évolutions ?	18
Quelle est la liaison avec les métadonnées du catalogue pour chaque couche ? (par la légende ? Possibilité de rechercher directement les MD dans l'interface ?)	19
Quels services web OGC peuvent être consultés dans l'interface ? Posent-ils des problèmes de performances ?	19
Est-il possible de sauvegarder et exporter des dessins ?	19
Existe-t-il un outil de production collaborative de données ?	19

Existe-t-il un module de téléversement de données ? Comment se fait le stockage (fichiers plats, injection en base) ? Qui a accès à ces données ?	19
Possibilité de sauvegarder/charger un contexte (créés par l'admin ou par des utilisateurs) ? [contexte cartographique au sens OGC]	19
Outil de visualisation cartographique : backOffice	20
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	20
Est-il possible de créer ou d'activer des webservices (flux interopérables, WMS ou WFS)	20
Est-il possible de décliner l'application cartographiques pour créer des visualiseurs thématiques ?	20
Comment se fait le paramétrage de nouveaux formulaires de requête et leur implémentation ? Est-elle facile ? Peut-on gérer les droits sur ces formulaires ? (par exemple, créer des masques de recherche métier liés à une couche particulière de la base)	20
L'application produit-elle du geoRSS ?	20
Demandes d'extraction : FrontOffice	20
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	20
Date de première version	20
Nombre de versions parues	20
Nombre d'utilisateurs / installations	21
Quel pilotage des évolutions ?	21
Quel financement des évolutions ?	21
Comment sont gérées les demandes d'extraction : par donnée, en une seule fois, par gestion d'un panier ?	21
L'outil d'extraction est-il accessible rapidement depuis tous les modules (catalogue, visualiseur, CMS)	21
Les demandes d'extraction se font par périmètre dessiné ou par sélection de communes ?	21
Quel est le découpage des données téléchargées ? (rectangle d'emprise, découpage par le polygone de demande ?)	21
Y a-t-il gestion d'un mailing lors de demandes d'extraction ? À quel moment et pour quels destinataires ? Y a-t-il calcul d'une estimation de temps avant la mise à disponibilité des fichiers ?	21
Les données peuvent être récupérées FTP, sur une URL provisoire ou bien le logiciel gère l'envoi sur support externe ?	21
Demandes d'extraction : BackOffice	21

Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	21
Quelle est la gestion des files d'attentes (validation / invalidation de demandes / changement des empilements) ?	22
Est-il possible d'accéder aux statistiques de charge serveur pour anticiper des blocages ?	22
Des packages déjà extraits sont-ils proposés de manière préférentielle en fonction des demandes ?	22
Quel est le format de stockage des projections ? Y a-t-il implémentation des bibliothèques IGNF ? Est-il possible d'ajouter aisément des projections/formats ?	22
<b>Catalogue</b>	22
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	22
Date de première version	22
Nombre de versions parues	22
Nombre d'utilisateurs / installations	22
Quel pilotage des évolutions ?	22
Quel financement des évolutions ?	22
Quelle est la liaison avec le portail ?	22
Quelle est la liaison entre une fiche de métadonnées et l'affichage de la donnée elle-même (dans le visualiseur) ?	22
Quelle est la liaison entre une fiche de métadonnées et la documentation présente dans le CMS ? (carte, document ressource...)	23
Peut-on visualiser les statistiques de consultation et quelle est la liaison avec les statistiques globales du site ou du portail ?	23
Quelle est la liaison avec les demandes de téléchargement (ou le panier) ?	23
<b>Serveur interopérable</b>	23
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	23
Date de première version	23
Nombre de versions parues	23
Nombre d'utilisateurs / installations	23
Quel pilotage des évolutions ?	23
Quel financement des évolutions ?	23
L'application produit-elle du WMS (quelles versions) ?	23
L'application produit-elle du WFS (quelles versions) ?	23
À la création d'un flux, comment se fait la création de la fiche de métadonnée associée ?	23

L'application produit-elle du WMS-C ?	24
L'application produit-elle du WMTS ?	24
L'application produit-elle du WCS ?	24
L'application produit-elle du WPS ?	24
L'application produit-elle du WFS-T ?	24
<b>Fonctions transverses</b>	<b>24</b>
Comment estimer la robustesse des composants ?	24
Quelle est l'homogénéité des environnements de développement / framework / langages ?	24
Comment estimez-vous, globalement, la facilité d'administration ?	24
Comment estimez-vous la force de la communauté (réactivité en cas de demandes liées à des blocages, nombre de contributeurs, hétérogénéité des contributeurs ?	24
SGBD : quelle puissance en cas de requêtes complexes ?	24
SGBD : quelle est l'intégration des fonctions géographiques de la plate-forme ? (les fonctions géographiques du SGBD sont-elles implémentées nativement ou bien le moteur cartographique utilisé déploie-t-il ses propres fonctions ?)	24
Quel est le niveau d'intégration des différents modules dans le portail, surtout en administration ?	25
Prise en main / courbe d'apprentissage	25
Sécurité : quelle authentification des utilisateurs ? (CAS, LDAP...)	25
Sécurité : quelle sécurisation des échanges avec l'internaute sur les données sensibles ?	25
Sécurité : quelle réactivité de l'éditeur en cas de failles de sécurité et mises à jour ?	25
Quelle fréquence de mise à jour, quelle facilité de déploiement ?	25
<b>Liste des fonctionnalités</b>	<b>25</b>
Visualiseur	25
Catalogue	26
Extracteur	26
Geopublisher	26
<b>Installation</b>	<b>26</b>
Guide d'installation	26
Préconditions	26
Configuration	27
Construction	28

Déploiement	28
Informations techniques	29
Mécanisme de déploiement	29
Informations techniques diverses	30
Java SSL, Keystores and Truststores	30
Procédure post-installation	31
Configuration des référentiels	31
Mapfishapp	31
Personnalisation de la liste des WMS	31
Personnalisation des couches chargées au démarrage	31
Préocédure d'installation du module search_address	31
Dépendances	31
Installation	31
Exécution dans Apache	32
Gestion des droits et sécurité	32
Introduction	32
Gestion	33
Catalogue	33
Principes	33
Interface	33
Serveur cartographique	33
Notion de groupe	34
Notion de région	34
Guía de instalación	35
Precondiciones	35
Configuración	36
Construcción	36
Despliegue	37
Informaciones técnicas	37
Mecanismo de despliegue	37
Algunas informaciones técnicas	38
Java SSL, Keystores and Truststores	39
Installation Guide	39
Preconditions	39
Configuration	40
Building all modules	40
Building submodules	41
Deploy	41

Technical Information	42
Deployment Mechanism	42
Random technical information	43
Java SSL, Keystores and Truststores	43
Utilisation	43
Administrateur	44
Documentation administrateur	44
Manuel utilisateur de GeoServer	44
Débuter	44
Démarrage rapide de l'interface d'administration web	44
Authentification	44
Serveur	44
Services	49
Données	49
Démos	51
Prévisualisation des couches	52
Ajouter un Shapefile	53
Commencer	53
Créer un nouvel Workspace	53
Créer un Store	55
Configuration de couche	55
Prévisualiser la couche	57
Ajouter une table PostGIS	59
Commencer	60
Créer un nouveau store	60
Configuration de la couche	63
Prévisualiser la couche	65
Styler une carte	66
Débuter	67
Créer un nouveau style	67
Interface Web d'administration	67
Interface basique	67
Page de bienvenue	67
Serveur	68
Status	68
Status Field Descriptions	69
Timestamps Field Descriptions	70
Information sur le contact	70

Champs d'information du contact	71
Global Settings	72
Global Setting Fields	73
Paramétrage de GeoWebCache	74
Enable direct WMS integration	75
Disk quota	76
Links	77
JAI	77
Memory & Tiling	78
Services	78
WCS	79
Service Metadata	79
WFS	80
Service Metadata	80
Features	80
Service Levels	81
GML	81
WMS	82
Service Metadata	82
Raster Rendering Options	82
Watermark Settings	82
SVG Options	84
Données	84
Espace de travail	85
Éditer un espace de travail	86
Ajouter ou supprimer un espace de travail	86
Entrepôt	87
Éditer un entrepôt	88
Ajouter un entrepôt	89
Supprimer un entrepôt	91
Couches	93
Types de couches	93
Éditez les données d'une couche	93
Info basique	94
Systems de référence des coordonnées	95
Boîtes englobantes	95
Paramètres de coverage (Raster)	95
Détail des Feature Type (Vecteur)	96

Éditez les informations de publication	96
Ajouter un supprimer une couche	98
Groupes de couches	101
Éditer un groupe de couches	101
Ajouter un groupe de couches	103
Supprimer un groupe de couches	105
Styles	105
Éditer des styles	106
Ajouter un style	109
Supprimer un style	109
Démos	110
Requêtes de démo	110
SRS	113
Prévisualisation de couches	115
Formats de sortie	116
Sortie image	117
Sortie texte	118
Sortie de données	119
Symbolisation	119
Introduction à SLD	119
Types de symbolisation	120
Métadonnées de style	120
GeoServer et SLD	120
Définitions	120
Symbolizer	120
Rule	120
FeatureTypeStyle	120
Un style simple	120
Dépannage	122
Livre de recettes SLD	122
Points	122
Exemple de couche ponctuelle	123
Point simple	123
Code	124
Details	124
Exemple simple avec bord	125
Code	125
Détails	126

Carré avec rotation	126
Code	126
Détails	127
Triangle transparent	127
Code	127
Détails	128
Point graphique	128
Code	129
Détails	129
Point avec étiquette par défaut	130
Code	130
Détails	131
Point avec étiquette stylisée	131
Code	132
Détails	133
Point avec rotation de l'étiquette	133
Code	134
Détails	135
Points basés sur les attributs	135
Code	136
Détails	137
Points basés sur le zoom	138
Code	141
Détails	141
Lines	142
Exemple de couche ligne	143
Ligne simple	144
Code	144
Détails	144
Ligne avec bord	145
Code	145
Détails	146
Ligne pointillée	146
Code	147
Détails	147
Voie ferrée (hachures)	147
Code	148
Détails	149

Symboles graphiques espacés	149
Code	150
Détails	150
Alternner les symboles avec dashoffset	150
Code	151
Détails	152
Line with default label	152
Code	153
Details	153
Label following line	154
Code	154
Details	155
Optimized label placement	155
Code	156
Details	157
Optimized and styled label	157
Code	158
Details	159
Attribute-based line	159
Code	159
Details	161
Zoom-based line	161
Code	164
Details	165
Polygons	166
Example polygons layer	166
Simple polygon	166
Code	167
Details	167
Simple polygon with stroke	168
Code	168
Details	169
Transparent polygon	169
Code	169
Details	170
Graphic fill	170
Code	171
Details	171

Hatching fill	171
Code	172
Details	173
Polygon with default label	173
Code	173
Details	174
Label halo	174
Code	175
Details	176
Polygon with styled label	176
Code	177
Details	178
Attribute-based polygon	178
Code	179
Details	180
Zoom-based polygon	181
Code	183
Details	184
Rasters	185
Exemple de raster	186
Gradient bicolore	186
Code	187
Détails	187
Gradient transparent	187
Code	188
Détails	188
Luminosité et contraste	188
Code	189
Détails	189
Gradient à trois couleurs	189
Code	190
Détails	190
Canal alpha	190
Code	191
Détails	191
Couleurs discrètes	191
Code	192
Détails	192

Gradient multicolore	193
Code	193
Détails	193
SLD Reference	194
PointSymbolizer	194
Syntax	194
Example	195
LineSymbolizer	195
Syntax	196
PolygonSymbolizer	197
Syntax	197
Example	197
Raster Symbolizer	198
Introduction	198
Elements and Syntax	198
Opacity	198
ColorMap	198
ChannelSelection	202
ContrastEnhancement	204
ShadedRelief	205
OverlapBehavior	205
ImageOutline	206
TextSymbolizer	206
Syntax	206
Example	207
Labeling	207
Controlling Label Placement	207
Basic SLD Placement	207
Code	208
PointPlacement	208
AnchorPoint	208
Displacement	209
Rotation	210
LinePlacement	210
Composing labels from multiple attributes	211
Geoserver Specific Enhanced Options	212
Priority Labeling (<Priority>)	212
Grouping Geometries (<VendorOption name="group">)	213

Overlapping and Separating Labels (<VendorOption name="spaceAround">)	214
followLine	215
maxDisplacement	215
repeat	215
labelAllGroup	215
maxAngleDelta	215
autoWrap	216
forceLeftToRight	216
conflictResolution	216
Goodness of Fit	216
Polygon alignment	216
Filters	217
Attribute filters	217
Spatial filters	217
Logical filters	218
Rules	218
Scale	220
SLD Extensions in GeoServer	220
Transformations de géométries en SLD	220
Extraction des sommets	220
Points début et fin	221
Ombre portée	222
Autres possibilités	223
Ajouter de nouvelles transformations	223
Point symbology in GeoServer	223
Mark support in GeoServer	223
Built-in symbols	223
The shape symbols	224
The TTF marks	224
Adding your own	225
External graphics in GeoServer	225
Dynamic symbolizers	226
Parameter substitution in SLD	227
A working example	227
Specifying symbolizers sizes in ground units	228
Label Obstacles	230
SLD Tips and Tricks	232

Dealing with mixed geometry types	232
Restructuring the data	232
Split the table	232
Separate geometry columns	232
Add a geometry type column	232
Create views	233
Using non-standard SLD functions	233
Utilisateur	234
Documentation utilisateur	234
Visualiseur – mode d'emploi	234
Introduction	235
Ergonomie générale de l'interface	235
Barre d'outils zoom-mesure-impression	235
Changement d'échelle	236
Présentation des couches	237
Ajout une couche d'un serveur OGC distant	237
Styler une couche	240
Requêter une couche	241
Editeur – mode d'emploi	241
Introduction	241
Ergonomie générale de l'interface	241
Charger une couche editable	241
Modification d'un objet	242
Saisie d'un objet	242
Catalogue – mode d'emploi	243
Introduction	243
Création d'une fiche	243
Saisie d'une fiche	244
Import d'une fiche	252
Administration d'une fiche	253
Mise à jour d'une fiche de métadonnée et de la donnée	255
Mise à jour d'une donnée issue d'une métadonnée de la V1	255
F.A.Q.	256
F.A.Q.	256
Gouvernance	256
À quoi sert le PSC (Project Steering Committee) ?	256
Comment contacter le PSC (Project Steering Committee) ?	256
Comment obtenir un accès à la forge ?	256

Technique	257
Quels sont les standards implémentés dans geOrchestra ?	257
Quelle est cette limite de 114 Ko qui apparait parfois dans mapfishapp ?	257
La recherche par adresse ne fonctionne pas ...	257
Comment brancher la recherche référentiels ?	258
En mode édition, la page ralenti fortement, j'ai un message d'erreur	258
Comment étendre les types de format possibles dans l'extracteur ?	258
Impossible d'ajouter une couche raster (ecw, jpg2000, ...)	258
Communauté	258
Listes de diffusion	258
Qui développe geOrchestra ?	258
Qui utilise geOrchestra ?	259
Télécharger geOrchestra	259
Release	259
Code source	259
Packages	259
Démonstration	259
À propos	260
Partagez vos données avec vos partenaires	260
Rejoignez la communauté	260

## Fonctionnalités principales



Le projet geOrchestra a pour objectif de développer une Infrastructure de Données Spatiales **modulaire, interopérable et libre**, en se basant sur les meilleurs composants cartographiques disponibles.

Référencé par l'[Adullact](#), geOrchestra propose une solution clef en main pour publier et partager vos données localisées sur intranet et internet.

The screenshot shows the geOrchestra website's homepage. The header is purple with the project name. The main content area contains several sections of text and small images illustrating the software's features. A sidebar on the left lists various links related to the project, such as 'Réseau d'interactions', 'Projets pilotes', 'Géoservices', 'Géodonneurs', 'Géovisualisation', 'Géoéditeur', 'Géodata', 'Géofonctionnalités', 'Géoweb', 'Géodistribution', 'Géodéveloppement', and 'Géodéveloppement'. The footer at the bottom right includes a link to 'Liens utiles'.

Pour une présentation visuelle de la solution geOrchestra et de ses composants, nous vous invitons à consulter cet [\*Aperçu des fonctionnalités\*](#). Vous pouvez également regarder la [\*Fiche d'identité du projet geOrchestra\*](#) pour avoir une connaissance des fonctionnalités offertes dans geOrchestra.

Vous pouvez trouver de l'aide grâce à la liste de diffusion communautaire : voir la section Communauté.

Enfin tenez vous au courant des évolutions du projet via [le blog](#) et le compte [Twitter](#) dédié.

## Fonctionnalités principales

Le projet geOrchestra est bâti autour d'une suite de modules complètement indépendants et interopérables :

- un catalogue, muni d'une fonction permettant de géo-publier des couches cartographiques,
- un visualiseur et éditeur,
- un extracteur,
- un serveur cartographique,
- un site éditorial.

Ces différents modules sont faiblement couplés, ce qui signifie que le fonctionnement de chaque module n'affecte pas les autres. La communication entre modules utilise les standards de l'OGC.

## Une réponse efficace à la directive INSPIRE



geOrchestra a été conçu pour répondre aux exigences de la [directive INSPIRE](#), qui vise à établir une infrastructure de données géographiques à l'échelle Européenne, dans l'objectif de répondre aux enjeux du développement durable :

- services de découverte, de visualisation et de téléchargement des données géographiques,
- lien permanent entre la donnée géographique et sa métadonnée,
- respect des normes et standards prônés par INSPIRE (cf. ci-dessous),
- outils d'administration facilitant la prise en compte d'INSPIRE (thèmes INSPIRE pré-enregistrés, thesaurus gemet, évaluation de la conformité INSPIRE, etc),
- une réutilisation facilitée des données grâce à la mise à disposition de services web.

## Implémentation des standards

geOrchestra repose sur les standards de l'OGC, notamment :

- [WMS](#) : Web Map Service
- [WMTS](#) : Web Map Tile Service
- [WFS](#) : Web Feature Service (dont transactionnel)
- [WCS](#) : Web Coverage Service
- [WMC](#) : Web Map Context
- [SLD](#) : Styled Layer Descriptor
- [FE](#) : Filter Encoding
- [CSW](#) : Catalog Service

## Une communauté d'utilisateurs et de développeurs

Le projet est supporté par une communauté croissante de développeurs et d'utilisateurs enthousiastes.

Plusieurs sociétés fournissent également un support commercial :

- [Camptocamp](#) : société de service en logiciel libre, basée à Chambéry, est concepteur de la solution dans le cadre du projet [GeoBretagne](#). Camptocamp propose des offres d'intégration de l'IDS au sein de votre infrastructure, du support, mais également des prestations de développement sur mesure et des formations (utilisateurs, administrateur système et administrateur de données).
- [DotGee](#) : société basée à Rennes, offre également des prestations de support et développement sur mesure autour de l'IDS.

## Documentation

## Une réponse efficace à la directive INSPIRE

Cette section rassemble la documentation autour du projet geOrchestra. Vous pouvez télécharger la documentation en un seul fichier PDF : [georchestra.pdf](#)

## Gouvernance

### Comité de Pilotage du Projet (PSC)

#### Introduction

##### Qu'est-ce que geOrchestra ?

geOrchestra est une Infrastructure de Données Spatiales libre, modulaire, interopérable. Elle intègre les logiciels libres GeoServer, GeoNetwork, GeoWebCache, OpenLayers, GeoExt, Mapfish et s'appuie sur les services web standardisés par l'OGC pour offrir à ses utilisateurs un ensemble de fonctionnalités avancées.

##### La philosophie de développement

- urbaniser le système et respecter une architecture modulaire ;
- Respecter au maximum les standards existants (dont OGC), tant sur les interfaces externes qu'internes (communications entre modules) ;
- Travailler avec les communautés libres existantes, c'est à dire : privilégier la contribution à ces projets et éviter les forks ;
- rendre publiques très tôt les sources : Forge redmine et Adullact (dépôt de sources, démos publiques, suivi ouvert des bugs, forums, etc.) ;
- s'entourer des conseils de partenaires (Adullact) et de personnes techniques (développeurs) ;
- méthode itérative de développements (AGILE).

##### Responsabilité du PSC

Ce document décrit la façon dont le comité de pilotage (PSC, *Project Steering Committee*) de geOrchestra détermine l'appartenance de ses membres, et prend des décisions sur tous les aspects du projet geOrchestra - à la fois techniques et non techniques.

Exemples des responsabilités dévolues au PSC sans que cela soit exhaustif :

- conduite de la roadmap ;
- élaboration de normes techniques et politiques (par exemple les normes de codage, de nommage des fichiers conventions, etc ..) ;
- prévoir les mises à jour régulières du logiciel geOrchestra ;
- examen des RFC d'amélioration technique du logiciel ;
- projet d'infrastructure (par exemple, CVS / SVN options, Bugzilla, hébergement, etc ..) ;
- formalisation d'affiliation avec des entités extérieures telles que l'OSGeo ;
- fixer les priorités du projet, notamment en matière de projet de parrainage ;
- la création et la supervision des sous-commissions spécialisées (par exemple des projets d'infrastructure, formation) ;
- vote des droits de commits sur proposition de la communauté des développeurs.

En bref, les votes sur les propositions de l'équipe de projet ont lieu sur la liste dévolue aux développeurs (voir page [Communauté](#)). Les propositions sont disponibles pour examen

## Une réponse efficace à la directive INSPIRE

pendant au moins deux jours ouvrables. Puis le vote a lieu et s'il n'y a pas de rejet, la motion est considérée comme acceptée. Pour plus d'informations, voir la page [Gestion des RFC](#).

### **Observations**

Le président est l'arbitre ultime, en cas de désaccord. La règle de la majorité absolue peut être utilisée pour outrepasser un veto d'obstruction, mais il est prévu que dans des conditions normales, les auteurs de veto doivent être convaincus de façon argumentée de retirer leur droit de veto. Le consensus est privilégié.

D'autres «comités» distincts pourront être mis en place pour gérer des conférences, la documentation et les sites web. Cela dit, il est prévu que le PSC soit en grande partie responsable de la création de ces comités.

### **Membres du PSC**

Le PSC est composé d'individus contributeurs techniques (c'est-à-dire des développeurs) et des membres éminents de la communauté des utilisateurs de geOrchestra. Il n'y a pas de nombre déterminé des membres du PSC, bien que le désir initial est de fonctionner à 8 membres. Les catégories des membres doivent être équilibrées (utilisateurs/développeurs).

### **Ajout de membres au PSC**

Tout membre de la liste de diffusion dévolue aux développeurs (voir page [Communauté](#)) peut, à tout moment, proposer quelqu'un en tant que nouveau membre du PSC. Seuls les membres actuels du PSC peuvent voter sur les nouveaux membres. Le candidat doit recevoir un vote à la majorité des membres existants avant d'être ajouté au PSC.

### **Retrait d'un membre du PSC**

Si pour une raison quelconque un membre du PSC n'est pas en mesure de participer pleinement au comité, il est évidemment libre de se retirer. Si un membre n'est pas actif (par exemple, pas de vote ou de participation par courrier électronique) pendant une période de deux mois, le PSC se réserve le droit de présenter des candidatures pour le remplacer. Si cette personne redevient active, elle sera la bienvenue, mais nécessitera une nouvelle candidature.

### **Responsabilités des membres du PSC**

#### **Conduite des développements**

Les membres du PSC peuvent prendre un rôle actif dans la conduite du développement de nouvelles fonctions dans leurs centres d'intérêt. Une fois qu'une évolution a été acceptée et que le feu vert a été donné pour procéder à son intégration au trunk, cela ne veut pas dire que les membres sont libérés de leurs obligations. Les membres du PSC ayant voté "+1" pour une évolution ont l'engagement moral de rester engagés en s'assurant que l'évolution a été mise en œuvre et correctement documentée. Notez que cela s'applique non seulement aux demandes de modification qui influent sur le code, mais aussi celles qui affectent le site web, l'infrastructure technique, les politiques et les normes.

#### **Participation aux réunions**

Les membres du PSC sont censés participer à des réunions pré-programmées de développement sur IRC ou par audio-conférence. Si un membre du PSC ne peut être présent à une réunion, il doit en informer par mail l'organisateur de la réunion.

#### **Participation aux listes de diffusion**

## Une réponse efficace à la directive INSPIRE

Les membres du PSC sont censés être actifs sur les listes de diffusion (voir [Communauté](#)). Les membres non-développeurs du PSC ne sont pas tant là pour répondre à des questions au niveau du code sur la liste de diffusion des développeurs, mais plutôt pour leurs réflexions et opinions sur les exigences du niveau de l'utilisateur et sur les problèmes de compatibilité lors de discussions sur les standards.

Fabrice Phung est déclaré président initial du PSC.

Les membres fondateurs sont :

- Lydie Vinsonneau
- Fabrice Phung
- Yves Jacolin

Les membres suivants ont été ajoutés à la PSC, après discussion et consensus : \* François Van Der Biest \* Pierre Mauduit

Mise à jour : 06/06/2011

### **Gestion des RFC**

Cette section détaille le fonctionnement des RFC (*Request For Comment*). Les RFC sont des propositions formalisées pour proposer des modifications majeures au projet geOrchestra.

Exemples de modifications qui font l'objet de RFC (voir plus bas pour une liste plus précise) :

- fonctionnalités majeures ;
- ré-architecture du code ;
- améliorations du processus communautaire ;
- propriété intellectuelle.

### **Comment cela fonctionne**

Le cycle de vie d'une RFC est celle-ci :

1. le développeur a l'intention de réaliser une modification importante ;
2. il communique sur la liste (voir [Communauté](#)) avec la communauté sur les modifications ; tout le monde peut commenter les propositions sur la liste, mais seuls les voix des membres du PSC seront comptées.
3. le développeur réalise les modifications du code ;
4. il écrit une RFC pour détailler ses modifications et ses conséquences ;
5. le PSC vote la RFC. Les propositions doivent être disponibles pour examen pendant au moins deux jours ouvrables avant qu'une décision finale puisse être faite.
6. si la RFC est acceptée, le développeur commite le code ou réalise le patch nécessaire et rédige ou modifie la documentation.

### **Vote d'une RFC**

Le PSC a la charge de voter les RFC proposées par la communauté. Le processus de vote est le suivant :

- chaque membre du PSC possède un vote qui peut être -1 (rejet), -0, +0, +1 (acceptation) ;
- un membre du PSC qui vote négativement doit fournir un minimum d'explication ;

## Une réponse efficace à la directive INSPIRE

- un membre du PSC qui vote négativement doit proposer une solution de rechange en un temps limité ;
- un vote de -0 indique un désaccord doux, mais n'a aucun effet. Le 0 indique l'absence d'avis. Un +0 indiquent un appui doux, mais n'a aucun effet ;
- l'auteur de la RFC doit rédiger un minimum de retour dans sa proposition ;
- un vote négatif est analysé pour voir comment modifier le critère bloquant pour obtenir un vote positif.
- la proposition est considérée comme acceptée après qu'une majorité absolue a voté +1 et toutes les remarques sur les votes négatifs ont été résolues. L'auteur de la proposition doit annoncer si celle-ci est acceptée (proposition acceptée par la majorité des committers) ou s'il retire sa proposition (veto).

Le président dispose d'un vote. Il est chargé de maintenir à jour la liste des membres du PSC du projet. L'ajout et la suppression d'un membre au comité de pilotage, ainsi que la sélection d'un président doit être traitée comme une proposition au PSC.

Le président statue dans les cas de litiges sur un vote.

Une proposition ne sera acceptée que si elle reçoit +2 (y compris l'auteur) et pas de veto (-1).

Si un veto est opposé à une proposition, et que celle-ci ne peut être revue pour satisfaire l'ensemble des votants, alors la proposition peut être soumise à un nouveau vote dans lequel une majorité de tous les votants éligibles indiquant +1 est suffisante pour l'adopter. Notez que c'est la majorité de tous les membres du PSC, et pas seulement ceux qui ont effectivement voté.

## ***Implémentation d'une RFC***

Les RFC proposées, discutées et votées sont disponible dans la section [RFC dans le wiki](#) de la forge.

## ***Quand le vote est-il obligatoire ?***

Le vote est obligatoire dans les cas suivants :

- Toute modification de la composition du comité (nouveaux membres, suppression de membres inactifs)
- Modifications au projet d'infrastructure (par exemple l'outil, l'emplacement ou la configuration de fond)
- Tout ce qui peut causer des problèmes de compatibilité descendante.
- Ajout des quantités importantes de nouveau code.
- Modification API inter-sous-système, ou des objets.
- Les questions de procédure.
- Lorsque une release doit avoir lieu.
- Tout en matière de relations avec des entités extérieures telles que l'OSGeo.
- Tout ce qui pourrait être sujet à controverse.

Les RFC proposées, discutées et votées sont disponibles dans la section [RFC dans le wiki](#) de la forge.

## ***Présentation***

Une réponse efficace à la directive INSPIRE

## Aperçu des fonctionnalités

Pour une liste des fonctionnalités plus exhaustive, allez sur la page [Liste des fonctionnalités](#).

## Architecture

### L'approche par modules

L'architecture de geOrchestra est composée de plusieurs modules autonomes. Une implémentation geOrchestra peut ainsi évoluer en douceur : commencer par le serveur cartographique, puis comporter un ou plusieurs visualiseurs... Jusqu'à former un ensemble complet.

### Le respect des normes

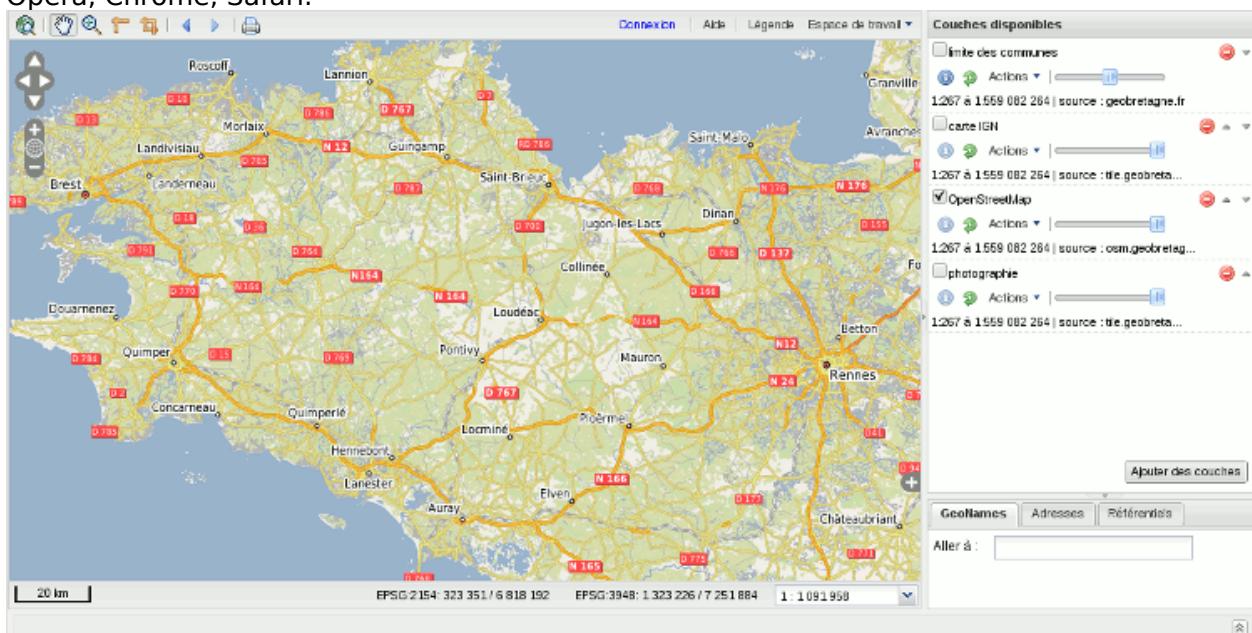
Les modules respectent les normes OGC préconisées par INSPIRE. Toutes les interactions entre modules utilisent ces normes : le système peut ainsi interopérer avec d'autres systèmes répondant aux mêmes normes, mais il est aussi lui-même interopérable en interne, et peut aisément s'adjointre les fonctions d'un composant externe. Vous pouvez par exemple utiliser QGIS ou ArcGIS pour fabriquer les symbolisations de cartes, puis exporter ce travail dans geOrchestra.

### Le suivi de la charge

Geoserver est utilisé dans des installations subissant de très fortes charges. Son intégration native du tuilage lui permet de répondre aux fortes sollicitations ou peut servir des couches très volumineuses.

## Visualiseur

Le visualiseur geOrchestra est un module autonome de consultation de cartes sur internet. Depuis un navigateur web, le visualiseur permet de rechercher, de charger, de consulter ces cartes. Le visualiseur fonctionne sur les versions récentes de Firefox, Internet Explorer, Opera, Chrome, Safari.



Depuis l'interface du visualiseur, l'utilisateur peut parcourir des catalogues de données respectant les normes INSPIRE, prendre connaissance du contenu des données (source, millésime, validité, droits...) en consultant les fiches associées, avant de les charger sur la

## Une réponse efficace à la directive INSPIRE

carte.

Il peut aussi interroger directement les serveurs "WMS" dont il connaît l'adresse, ou enfin utiliser les serveurs déjà référencés par le visualiseur.

L'interface est conçue pour travailler sur toutes tailles d'écran. Les panneaux latéraux repliables offrent une surface de visualisation maximale sur de petits écrans, alors que le chargement par tuiles optimise l'affichage sur de grands écrans.

### **Note**

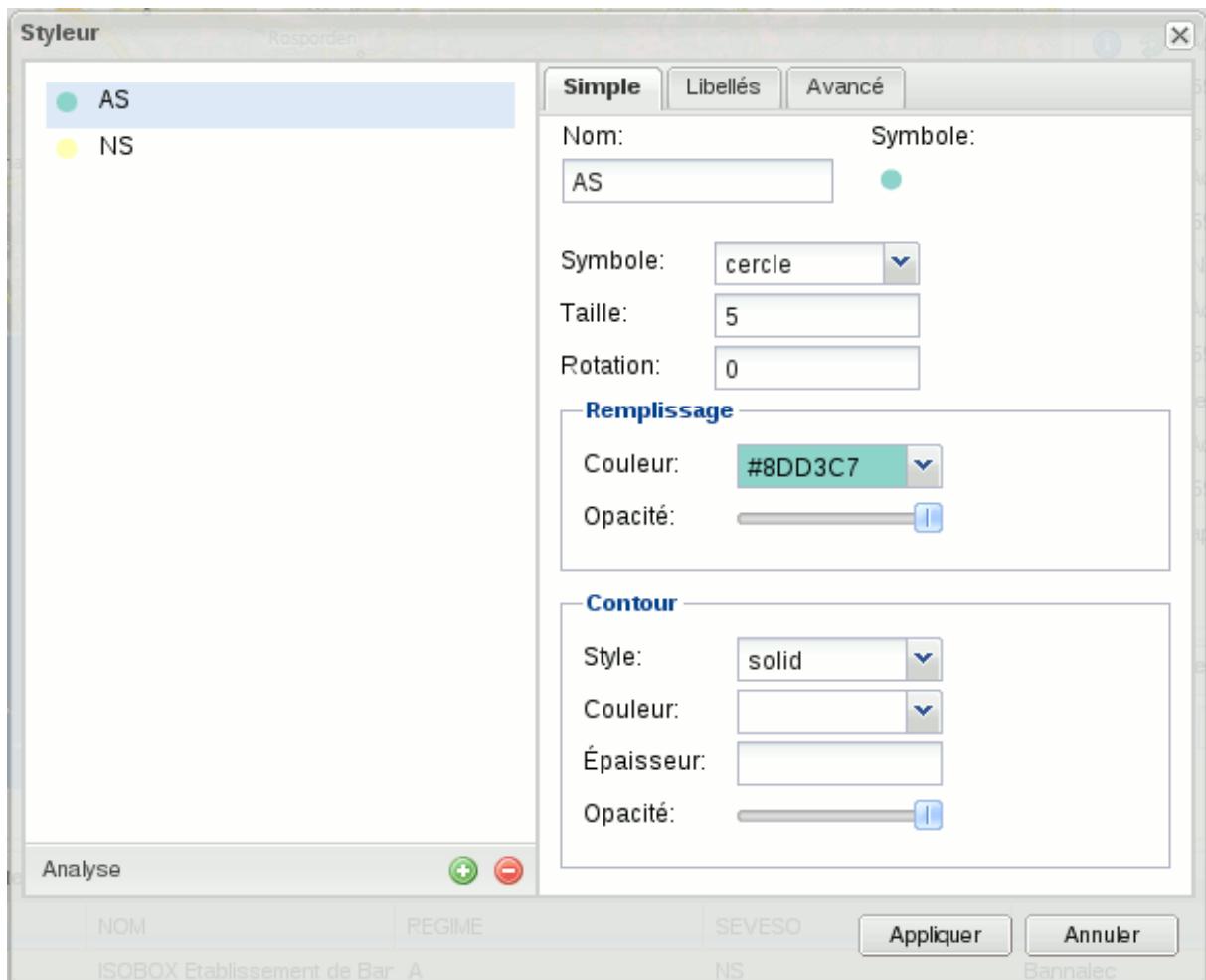
Les configurations de cartes sont enregistrées dans des documents très légers appelés "contextes" qui peuvent être rechargés, modifiés, échangés avec des tiers.

Le visualiseur peut être inséré dans un site existant. Sa configuration de démarrage est personnalisable par l'administrateur du site : données chargées par défaut, projection, emprise, catalogue et serveurs ciblés, performance des navigateurs...

### **Styleur**

Le styleur est une fonction avancée du visualiseur pour personnaliser les cartes selon les attentes précises de l'utilisateur. Il permet en effet d'établir des règles de représentation (forme, couleur, transparence, police) en fonction des données : par exemple, représenter les éléments de type "forêt" en vert et ne pas afficher ceux de type "urbanisé", ou mettre en relief les "parcelles de moins de 500m<sup>2</sup>". Le styleur ne télécharge aucune donnée : les serveurs font le travail. Il peut travailler sur des données très volumineuses comme le cadastre avec le même niveau de performance.

Une réponse efficace à la directive INSPIRE



### Note

Les styles fabriqués peuvent être conservés ou échangés avec des tiers. L'utilisateur peut ainsi construire sa propre bibliothèque de cartes correspondant à ses besoins.

### Requêteur

À la différence du styleur, le requêteur télécharge les données dans le navigateur pour les représenter, afficher tous les attributs dans un tableau et, si souhaité, les exporter vers un tableau. Le requêteur procède par filtres géographiques (par exemple : intersection avec points, lignes, polygones tracés par l'utilisateur) et attributaires (par exemple : valeur comprise entre deux nombres) que l'on modifie jusqu'à obtenir le résultat souhaité.

## Une réponse efficace à la directive INSPIRE

Réponse à la directive INSPIRE

Requête sur TAR

Correspondre à toutes ces conditions :

- Intersection avec
- DEP = 29
- VILLE = Bannalec

condition condition spatiale

Recherche Fermer

GeoNames Adresses Référentiels

Aller à :

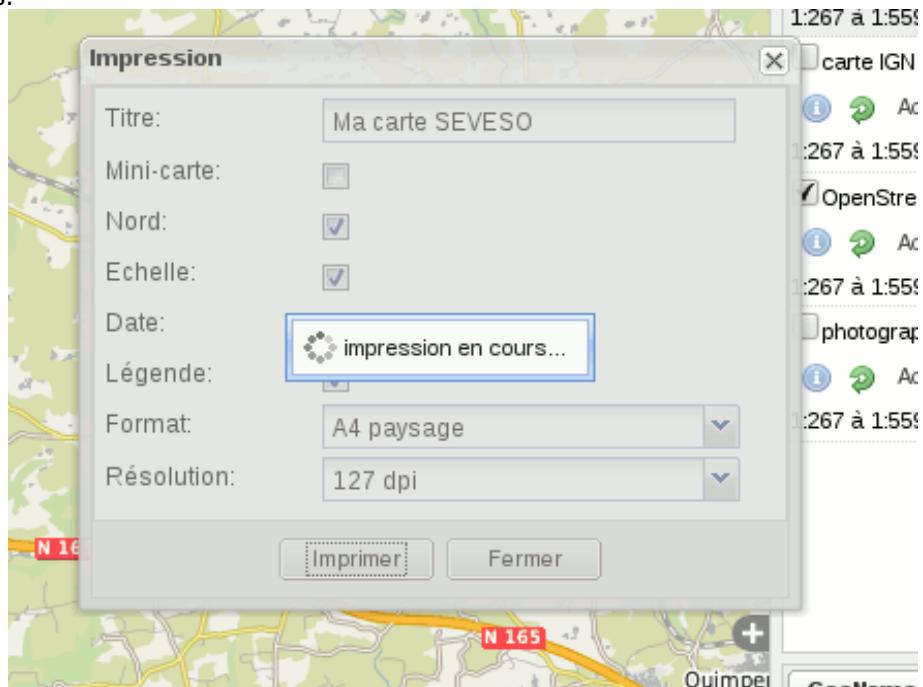
Résultats de requête

E CODE	NOM	REGIME	SEVESO	VILLE	DEP	COM
55-00506	ISOBOX Etablissement de Bar A	NS	NS	Bannalec	29	4

Effacer 1 résultat Zoom Export CSV

### Impression

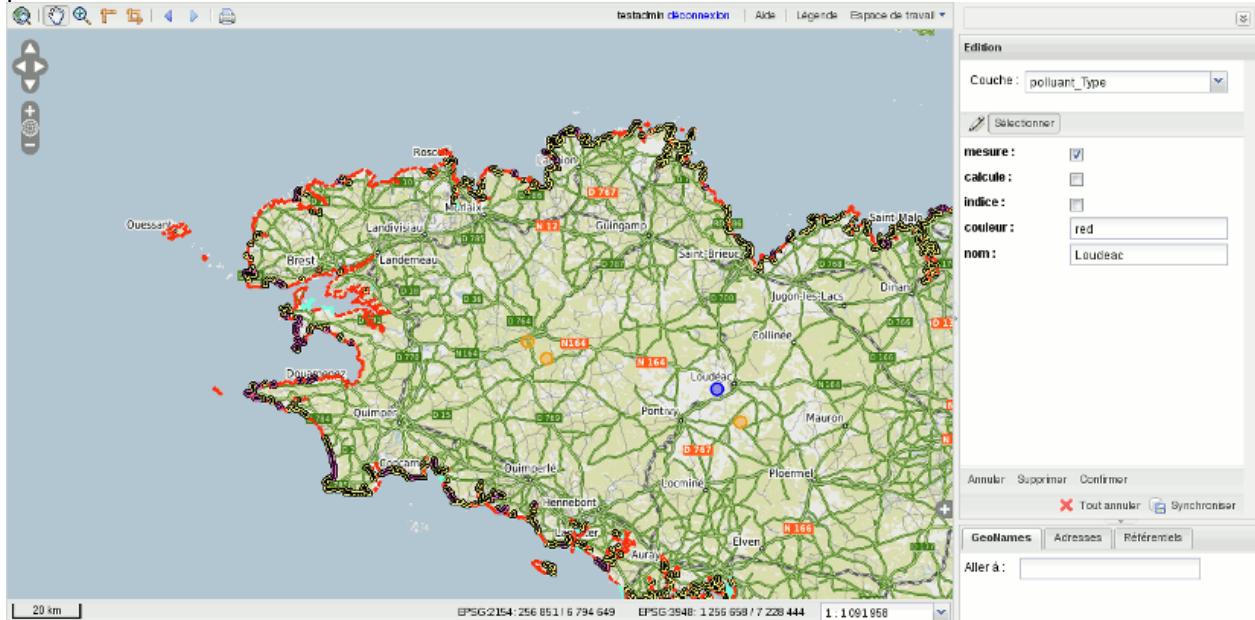
La fonction d'impression insère la carte dans un document PDF selon des gabarits préfabriqués.



### Éditeur

## Une réponse efficace à la directive INSPIRE

L'éditeur utilise la norme WFS-T (pour Transactionnel) pour permettre à l'utilisateur de modifier en direct les objets composant la carte. Il est possible de saisir et modifier des informations en ligne : un périmètre d'étude avec sa description, un document existant que l'on souhaite localiser, un tracé de projet... La fonction d'édition est une base de construction pour des cartes collaboratives.



## Catalogue

Le catalogue référence et décrit les données. Respectant les normes INSPIRE, le catalogue est un point d'entrée important par lequel l'utilisateur prend connaissance du patrimoine de données, recherche par thématique ou mots clefs les lots disponibles, et les soumet aux autres modules pour visualisation ou extraction. Le catalogue est basé sur Geonetwork, l'outil de référence pour la norme d'interrogation de catalogues CSW.

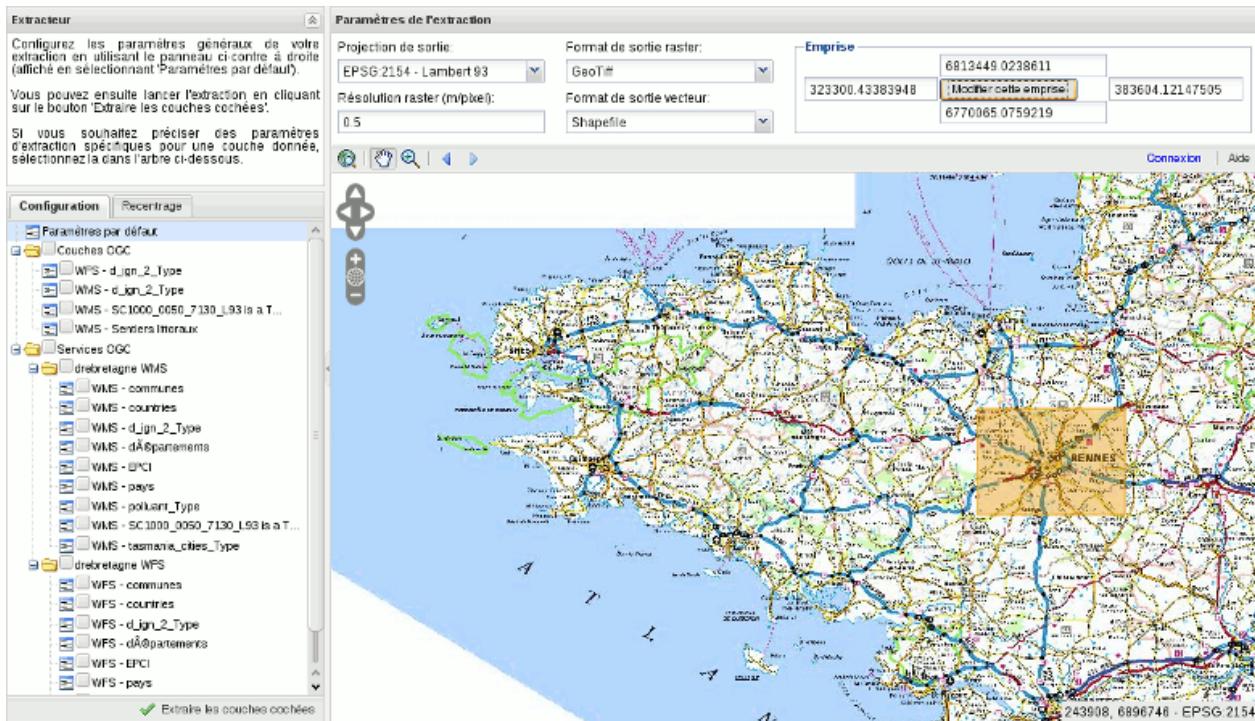
## Une réponse efficace à la directive INSPIRE

The screenshot displays two web-based interfaces side-by-side. On the left is the 'camp2camp' platform, featuring a search bar with fields for 'QUOI?' and 'OÙ?', a map view, and a sidebar with various dataset categories like 'Cartes & graphiques', 'Cartes statiques', etc. On the right is the 'geOrchestra' platform, which shows a search results page for 'TROUVEZ DES CARTES INTERACTIVES, DES JEUX DE DONNÉES, DES CARTES, DES IMAGES SATELLITES ...'. The results are categorized under 'COHÉRENCE LAND COVER', 'TRAITS LOGIQUES SOCIAUX PUBLICS 2005 PAR COMMUNE', 'PORTS DES COMMUNES ET BORDES', 'SITES INScritS SURFACIQUEs', and 'QUALITÉ DES EAUX DE Baignade'. Each result includes a summary, keywords ('Résumé', 'Mots clés'), and a preview icon labeled 'fichier d'aperçu'.

### Extracteur

L'extracteur est un module autonome permettant le téléchargement de données pour utilisation dans des logiciels tiers. En délimitant un secteur géographique, en choisissant une sélection de données, un format et une projection, on obtient un lot de données personnalisé. Cette opération étant souvent longue, l'extracteur émet un message à la fin du travail pour signaler que le téléchargement est prêt.

## Une réponse efficace à la directive INSPIRE



### Serveur Cartographique

Le serveur cartographique de geOrchestra est Geoserver, un puissant outil de publication aux normes INSPIRE. Equipé d'une interface graphique francisée, Geoserver offre à l'administrateur de données un contrôle fin sur tous les aspects de la publication : description des couches, formats de sortie, droits d'accès, styles... Si le format de prédilection des données est SHAPE, Geoserver sait composer avec de nombreuses autres sources de données et peut donc s'intégrer à des systèmes d'information existants.

Geoserver propose de plus un Web Processing Service, un service de traitement de données puissant aux capacités comparables aux systèmes d'information géographique.

### Geopublisher

Le geopublisher est un outil de productivité pour l'administrateur de données : il coordonne la description des données dans le catalogue avec leur publication sur les services de données. Le geopublisher permet une publication rapide des lots de données simples.

### Fiche d'identité du projet geOrchestra

<b>Portail public (CMS)</b>	<b>17</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	17
Date de première version	17
Nombre de versions parues	17
Nombre d'utilisateurs / installations	17
Quel pilotage des évolutions ?	17
Quel financement des évolutions ?	17
Le CMS comprend-il une gestion de groupes de travail ?	17

## Une réponse efficace à la directive INSPIRE

Quelle est la liaison avec le catalogue de métadonnées ? (affichage des dernières MD créées, des MD les plus consultées ...)	17
Y a-t-il localisation cartographique des contenus du CMS ?	17
<b>Administration du portail (CMS)</b>	<b>17</b>
Existe-t-il un module de gestion des conventions d'échange ?	17
Quelle est la granularité de gestion des droits ? (par contenus, par fonctionnalité, sur les fichiers...)	18
Comment est organisée la documentation (arborescence de fichiers ? Bases de données documentaire ?) et quelle liaison avec les éléments du CMS ?	18
Comment se fait la restitution des statistiques de consultation ?	18
<b>Outil de visualisation cartographique</b>	<b>18</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	18
Date de première version	18
Nombre de versions parues	18
Nombre d'utilisateurs / installations	18
Quel pilotage des évolutions ?	18
Quel financement des évolutions ?	18
Quelle est la liaison avec les métadonnées du catalogue pour chaque couche ? (par la légende ? Possibilité de rechercher directement les MD dans l'interface ?)	19
Quels services web OGC peuvent être consultés dans l'interface ? Posent-ils des problèmes de performances ?	19
Est-il possible de sauvegarder et exporter des dessins ?	19
Existe-t-il un outil de production collaborative de données ?	19
Existe-t-il un module de téléversement de données ? Comment se fait le stockage (fichiers plats, injection en base) ? Qui a accès à ces données ?	19
Possibilité de sauvegarder/charger un contexte (crées par l'admin ou par des utilisateurs) ? [contexte cartographique au sens OGC]	19
<b>Outil de visualisation cartographique : backOffice</b>	<b>20</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	20
Est-il possible de créer ou d'activer des webservices (flux interopérables, WMS ou WFS)	20
Est-il possible de décliner l'application cartographiques pour créer des visualiseurs thématiques ?	20
Comment se fait le paramétrage de nouveaux formulaires de requête et leur implémentation ? Est-elle facile ? Peut-on gérer les droits sur ces formulaires ? (par exemple, créer des masques de recherche métier liés à une couche particulière de la base)	20
L'application produit-elle du geoRSS ?	20

## Une réponse efficace à la directive INSPIRE

<b>Demandes d'extraction : FrontOffice</b>	<b>20</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	20
Date de première version	20
Nombre de versions parues	20
Nombre d'utilisateurs / installations	21
Quel pilotage des évolutions ?	21
Quel financement des évolutions ?	21
Comment sont gérées les demandes d'extraction : par donnée, en une seule fois, par gestion d'un panier ?	21
L'outil d'extraction est-il accessible rapidement depuis tous les modules (catalogue, visualiseur, CMS)	21
Les demandes d'extraction se font par périmètre dessiné ou par sélection de communes ?	21
Quel est le découpage des données téléchargées ? (rectangle d'emprise, découpage par le polygone de demande ?)	21
Y a-t-il gestion d'un mailing lors de demandes d'extraction ? À quel moment et pour quels destinataires ? Y a-t-il calcul d'une estimation de temps avant la mise à disponibilité des fichiers ?	21
Les données peuvent être récupérées FTP, sur une URL provisoire ou bien le logiciel gère l'envoi sur support externe ?	21
<b>Demandes d'extraction : BackOffice</b>	<b>21</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	21
Quelle est la gestion des files d'attentes (validation / invalidation de demandes / changement des empilements) ?	22
Est-il possible d'accéder aux statistiques de charge serveur pour anticiper des blocages ?	22
Des packages déjà extraits sont-ils proposés de manière préférentielle en fonction des demandes ?	22
Quel est le format de stockage des projections ? Y a-t-il implémentation des bibliothèques IGNF ? Est-il possible d'ajouter aisément des projections/formats ?	22
<b>Catalogue</b>	<b>22</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	22
Date de première version	22
Nombre de versions parues	22
Nombre d'utilisateurs / installations	22
Quel pilotage des évolutions ?	22
Quel financement des évolutions ?	22

## Une réponse efficace à la directive INSPIRE

Quelle est la liaison avec le portail ?	22
Quelle est la liaison entre une fiche de métadonnées et l'affichage de la donnée elle-même (dans le visualiseur) ?	22
Quelle est la liaison entre une fiche de métadonnées et la documentation présente dans le CMS ? (carte, document ressource...)	23
Peut-on visualiser les statistiques de consultation et quelle est la liaison avec les statistiques globales du site ou du portail ?	23
Quelle est la liaison avec les demandes de téléchargement (ou le panier) ?	23
<b>Serveur interopérable</b>	<b>23</b>
Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?	23
Date de première version	23
Nombre de versions parues	23
Nombre d'utilisateurs / installations	23
Quel pilotage des évolutions ?	23
Quel financement des évolutions ?	23
L'application produit-elle du WMS (quelles versions) ?	23
L'application produit-elle du WFS (quelles versions) ?	23
À la création d'un flux, comment se fait la création de la fiche de métadonnée associée ?	23
L'application produit-elle du WMS-C ?	24
L'application produit-elle du WMTS ?	24
L'application produit-elle du WCS ?	24
L'application produit-elle du WPS ?	24
L'application produit-elle du WFS-T ?	24
<b>Fonctions transverses</b>	<b>24</b>
Comment estimer la robustesse des composants ?	24
Quelle est l'homogénéité des environnements de développement / framework / langages ?	24
Comment estimez-vous, globalement, la facilité d'administration ?	24
Comment estimez-vous la force de la communauté (réactivité en cas de demandes liées à des blocages, nombre de contributeurs, hétérogénéité des contributeurs) ?	24
SGBD : quelle puissance en cas de requêtes complexes ?	24
SGBD : quelle est l'intégration des fonctions géographiques de la plate-forme ? (les fonctions géographiques du SGBD sont-elles implémentées nativement ou bien le moteur cartographique utilisé déploie-t-il ses propres fonctions ?)	24
Quel est le niveau d'intégration des différents modules dans le portail, surtout en administration ?	25

## Une réponse efficace à la directive INSPIRE

Prise en main / courbe d'apprentissage	25
Sécurité : quelle authentification des utilisateurs ? (CAS, LDAP...)	25
Sécurité : quelle sécurisation des échanges avec l'internaute sur les données sensibles ?	25
Sécurité : quelle réactivité de l'éditeur en cas de failles de sécurité et mises à jour ?	25
Quelle fréquence de mise à jour, quelle facilité de déploiement ?	25

### **Portail public (CMS)**

#### **Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Liferay, mais il n'y a pas de connexion forte entre le CMS et l'infrastructure de données spatiales. geOrchestra peut s'intégrer facilement dans tout site web.

#### **Date de première version**

La première version date de 2000.

#### **Nombre de versions parues**

Il y a eu 6 versions majeures.

#### **Nombre d'utilisateurs / installations**

Plusieurs millions.

#### **Quel pilotage des évolutions ?**

Classique : plusieurs sociétés contribuent à ce projet OpenSource sur la base de demandes client.

#### **Quel financement des évolutions ?**

Via société éditrice et partenaires.

#### **Le CMS comprend-il une gestion de groupes de travail ?**

Oui via la gestion des "pôles métier" qui comprennent partage de documents, d'événements, d'actualités et d'annuaires métier dédiés.

#### **Quelle est la liaison avec le catalogue de métadonnées ? (affichage des dernières MD créées, des MD les plus consultées ...)**

Via agrégateur de flux RSS : affichage des dernières MD créées & via recherche personnalisée

#### **Y a-t-il localisation cartographique des contenus du CMS ?**

Non, pas actuellement, mais cela est réalisable.

### **Administration du portail (CMS)**

#### **Existe-t-il un module de gestion des conventions d'échange ?**

Non.

Une réponse efficace à la directive INSPIRE

**Quelle est la granularité de gestion des droits ? (par contenus, par fonctionnalité, sur les fichiers...)**

3 profils existent :

- administrateur : droits sur la totalité des contenus,
- utilisateur : droits de proposer du contenu, validé ensuite par l'administrateur, et droit d'accès à du contenu non partagé,
- non loggué : droit de consultation grand public.

**Comment est organisée la documentation (arborescence de fichiers ? Bases de données documentaire ?) et quelle liaison avec les éléments du CMS ?**

La documentation du projet geOrchestra est disponible :

- dans une webapp "doc" déployée avec l'IDS pour ce qui est de la documentation utilisateur
- sur le site web georchestra.org pour ce qui est de la documentation administrateur

**Comment se fait la restitution des statistiques de consultation ?**

Possibilité d'inclure un tracker tel que Google Analytics. Auquel cas, la restitution se fait via l'interface dédiée.

**Outil de visualisation cartographique**

**Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Un développement spécifique a été réalisé car aucun visualiseur existant ne remplissait les fonctions demandées. Le visualiseur a été développé sur la base des technologies clientes OpenLayers, GeoExt, ExtJS et Spring, MapFish print pour la partie serveur.

Il s'interface (au sens de "communique") avec le serveur cartographique via les normes OGC WMS, WFS, SLD, FE.

**Date de première version**

2010

**Nombre de versions parues**

Il y a eu 24 "builds" à la date du 19 septembre 2011, cf <http://csm-bretagne.fr/hudson/job/georchestra/>

**Nombre d'utilisateurs / installations**

De l'ordre de la dizaine.

**Quel pilotage des évolutions ?**

Via le PSC, cf <http://www.georchestra.org/documentation/psc.html>

**Quel financement des évolutions ?**

Dans l'esprit le plus traditionnel du développement logiciel libre, c'est-à-dire, par le biais de projets financés mais également contributions externes bénévoles.

Une réponse efficace à la directive INSPIRE

**Quelle est la liaison avec les métadonnées du catalogue pour chaque couche ?  
(par la légende ? Possibilité de rechercher directement les MD dans l'interface ?)**

Au niveau de la légende de chaque couche, le visualiseur affiche un lien vers la fiche de métadonnées correspondante. Celle-ci s'ouvre dans une nouvelle fenêtre. L'utilisateur peut également rechercher dans le catalogue en utilisant les mots clés des métadonnées, tout en restant dans l'interface du visualiseur.

**Quels services web OGC peuvent être consultés dans l'interface ? Posent-ils des problèmes de performances ?**

Les services WMS, WMS-C & WFS peuvent être nativement consultés et requêtés dans le visualiseur.

Il y a cependant des limitations :

- les WMS-C sont consultables uniquement via chargement de fichiers de contexte (OGC WMC)
- les WFS sont soumis aux limites intrinsèques des navigateurs en terme de nombre d'objets et de vertex simultanément affichables (et cette limite varie fortement selon le navigateur choisi)

Pour passer outre ces limitations, il y a la possibilité de "proxifier" les couches WMS-C et WFS via le moteur cartographique de la plateforme (GeoServer), qui permettra de les afficher sous la forme de couches WMS, tout en conservant une très bonne ergonomie de consultation (notamment grâce au mode tuilé).

**Est-il possible de sauvegarder et exporter des dessins ?**

Non

**Existe-t-il un outil de production collaborative de données ?**

Oui, il s'agit d'un éditeur collaboratif basé sur le protocole OGC WFS-T. Cf <http://www.georchestra.org/documentation/utilisateur/edition.html> pour une présentation plus complète.

**Existe-t-il un module de téléversement de données ? Comment se fait le stockage (fichiers plats, injection en base) ? Qui a accès à ces données ?**

Non, pas dans le visualiseur. Oui, au niveau du catalogue : tout type de données (PDF, CSV, JPG, PNG ...) peut être "attaché" à une fiche de métadonnées. Le stockage est alors effectué dans un espace dédié sur le disque. L'accès à ces données peut être limité pour chaque fiche de métadonnées, sur la base du système de gestion des droits (organismes/rôles).

**Possibilité de sauvegarder/charger un contexte (créés par l'admin ou par des utilisateurs) ? [contexte cartographique au sens OGC]**

Dans le visualiseur, toute composition cartographique est OGC WMC :

- l'administrateur choisit le contexte par défaut du visualiseur sous la forme d'un fichier WMC,
- l'administrateur peut créer un certain nombre de contextes et proposer autant de liens vers des visualiseurs avec des thèmes distincts,
- tout utilisateur peut sauvegarder (sur son poste) sa composition cartographique au format WMC,
- tout utilisateur peut restaurer un contexte cartographique qu'on lui aura communiqué au format WMC,

Une réponse efficace à la directive INSPIRE

- tout utilisateur peut créer un permalien vers sa composition cartographique en appelant le visualiseur avec le paramètre ?wmc= suivi d'une URL vers le fichier WMC.

#### **Outil de visualisation cartographique : backOffice**

##### **Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Nous utilisons le moteur cartographique GeoServer 2, dont l'interface d'administration présente une très bonne ergonomie. Seuls les utilisateurs munis du rôle "ADMINISTRATOR" ont accès à cette interface. L'interface est intégrée à la suite d'applications via un bandeau reprenant les fonctionnalités de l'IDS (catalogue/visualiseur/extracteur/administration)

##### **Est-il possible de créer ou d'activer des webservices (flux interopérables, WMS ou WFS)**

Tout à fait, et sans avoir besoin de taper une seule ligne de commande : en ligne depuis l'interface de GeoServer. Chaque couche est activable/désactivable à volonté.

##### **Est-il possible de décliner l'application cartographiques pour créer des visualiseurs thématiques ?**

Oui, cf ci-dessus (contextes OGC WMC)

##### **Comment se fait le paramétrage de nouveaux formulaires de requête et leur implémentation ? Est-elle facile ? Peut-on gérer les droits sur ces formulaires ? (par exemple, créer des masques de recherche métier liés à une couche particulière de la base)**

Il n'y a pas de notion de formulaire de requête pré-existant : tout utilisateur est en mesure de créer sa propre requête métier sur toute couche WMS/WFS en utilisant des critères attributaires et géométriques.

#### **L'application produit-elle du geoRSS ?**

Oui, en utilisant la syntaxe suivante pour une couche nommée ns:test : <http://ids.monserveur.org/geoserver/wms/reflect?layers=ns:test&format=rss>

#### **Demandes d'extraction : FrontOffice**

##### **Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Un développement spécifique a été réalisé car aucun système d'extraction existant ne remplissait les fonctions demandées. L'extracteur a été développé sur la base des technologies clientes OpenLayers, GeoExt, ExtJS et Spring pour la partie serveur.

Il reçoit ses demandes de traitement par le contenu JSON qui lui est directement POSTé. Il est en mesure de traiter les couches WMS/WFS servies par le moteur cartographique local ou par des serveurs OGC distants.

#### **Date de première version**

2010, idem module visualiseur.

#### **Nombre de versions parues**

Idem module visualiseur.

Une réponse efficace à la directive INSPIRE

**Nombre d'utilisateurs / installations**

Probablement une petite dizaine.

**Quel pilotage des évolutions ?**

Idem module visualiseur.

**Quel financement des évolutions ?**

Idem module visualiseur.

**Comment sont gérées les demandes d'extraction : par donnée, en une seule fois, par gestion d'un panier ?**

Gestion d'un panier, traité comme un lot unique de plusieurs couches de données.

**L'outil d'extraction est-il accessible rapidement depuis tous les modules (catalogue, visualiseur, CMS)**

Il est accessible rapidement depuis le catalogue (lien via le menu "actions sur la sélection") et depuis le CMS (intégration directe). Le lien depuis le visualiseur sera implémenté au cours de l'année 2012 (cf ticket <http://csm-bretagne.fr/redmine/issues/2049>)

**Les demandes d'extraction se font par périmètre dessiné ou par sélection de communes ?**

Les deux modalités sont possibles.

Noter que, pour la sélection par communes, tout autre type d'objet polygonal est acceptable (ex: emprise de forêts), du moment que celui-ci est servi en WFS par GeoServer dans l'espace de nommage configuré. Noter enfin que le périmètre d'extraction peut être agrémenté d'un buffer configurable.

**Quel est le découpage des données téléchargées ? (rectangle d'emprise, découpage par le polygone de demande ?)**

Le découpage est effectué selon un rectangle qui peut être soit l'emprise native de la couche, soit une emprise définie manuellement, soit une emprise correspondant à un objet de la base de données, éventuellement augmenté d'un buffer.

**Y a-t-il gestion d'un mailing lors de demandes d'extraction ? À quel moment et pour quels destinataires ? Y a-t-il calcul d'une estimation de temps avant la mise à disponibilité des fichiers ?**

Oui, un email est envoyé lors de la prise en compte de la demande d'extraction et lors de la fin du traitement. Le destinataire est le demandeur, et éventuellement un administrateur de données. Il n'y a pas d'estimation de temps de traitement.

**Les données peuvent être récupérées FTP, sur une URL provisoire ou bien le logiciel gère l'envoi sur support externe ?**

Les données sont récupérées via une URL à usage limité dans le temps.

**Demandes d'extraction : BackOffice**

**Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Développement spécifique.

Une réponse efficace à la directive INSPIRE

**Quelle est la gestion des files d'attentes (validation / invalidation de demandes / changement des empilements) ?**

En cours de développement (sept 2011) : interface de gestion de la file d'attente, changement de priorité d'un job, arrêt d'un job. Les jobs sont auto-modérés par l'application, sur la base de critères (essentiellement de volumétrie) fixés à l'avance par l'administrateur de l'IDS.

**Est-il possible d'accéder aux statistiques de charge serveur pour anticiper des blocages ?**

Il n'y a pas de blocage possible : les demandes trop volumineuses sont refusées et l'utilisateur en est averti par courriel.

**Des packages déjà extraits sont-ils proposés de manière préférentielle en fonction des demandes ?**

Pas au niveau de l'extracteur, mais cela est possible via le module catalogue.

**Quel est le format de stockage des projections ? Y a-t-il implémentation des bibliothèques IGNF ? Est-il possible d'ajouter aisément des projections/formats ?**

Les projections supportées par l'extracteur sont gérées au niveau du module epsg-extension de l'IDS. Par défaut, un très grand nombre de projections sont supportées, mais il est toujours possible d'en ajouter (notamment celles de la bibliothèque IGNF).

**Catalogue**

**Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

GeoNetwork OpenSource. Interfaçage via flux OGC WMS/WFS/WCS & CSW.

**Date de première version**

2000

**Nombre de versions parues**

> 14 (cf <http://sourceforge.net/projects/geonetwork/files/GeoNetworkOpensource/>)

**Nombre d'utilisateurs / installations**

Plusieurs milliers.

**Quel pilotage des évolutions ?**

PSC, projet OSGeo, cf <http://trac.osgeo.org/geonetwork/wiki/PSC>

**Quel financement des évolutions ?**

Nations Unies et financeurs privés.

**Quelle est la liaison avec le portail ?**

Flux RSS pour une intégration fine & intégration via iframe pour consultation complète.

**Quelle est la liaison entre une fiche de métadonnées et l'affichage de la donnée elle-même (dans le visualiseur) ?**

Une réponse efficace à la directive INSPIRE

Via une liste déroulante, il est possible d'envoyer une sélection de fiches de métadonnées vers le visualiseur, afin de visualiser complètement les couches OGC référencées dans ces fiches de métadonnées.

**Quelle est la liaison entre une fiche de métadonnées et la documentation présente dans le CMS ? (carte, document ressource...)**

Une fiche de métadonnées peut lier vers toute page du CMS.

**Peut-on visualiser les statistiques de consultation et quelle est la liaison avec les statistiques globales du site ou du portail ?**

?

**Quelle est la liaison avec les demandes de téléchargement (ou le panier) ?**

Il est possible d'envoyer simultanément une ou plusieurs fiches de métadonnées vers l'extracteur, afin de télécharger les données référencées dans ces fiches (via services WMS et/ou WFS)

**Serveur interopérable**

**Quel outil a été retenu ? Comment s'interface-t-il avec les autres éléments de l'architecture ?**

Il s'agit de GeoServer 2, sur lequel reposent les flux WMS, WFS & WCS de la plateforme.

**Date de première version**

2001

**Nombre de versions parues**

Beaucoup ...

**Nombre d'utilisateurs / installations**

Plusieurs milliers

**Quel pilotage des évolutions ?**

Via le PSC, cf <http://docs.geoserver.org/latest/en/developer/policies/psc.html>

**Quel financement des évolutions ?**

Comme tout projet OpenSource supporté par plusieurs sociétés commerciales et une communauté bénévole.

**L'application produit-elle du WMS (quelles versions) ?**

Oui : 1.1.1 & 1.3.0

**L'application produit-elle du WFS (quelles versions) ?**

Oui : 1.0.0 & 1.1.0

**À la création d'un flux, comment se fait la création de la fiche de métadonnée associée ?**

C'est en fait le processus inverse qui est suivi dans geOrchestra : il y a la possibilité de créer une fiche de métadonnées dans le catalogue et, depuis cette même interface, d'uploader un

Une réponse efficace à la directive INSPIRE

zip contenant les données. Celles-ci seront alors utilisées par GeoServer pour produire une couche servie en WMS/WFS, automatiquement référencée dans la fiche de métadonnées.

**L'application produit-elle du WMS-C ?**

Oui, via GeoWebCache, intégré.

**L'application produit-elle du WMTS ?**

Oui, via GeoWebCache, intégré.

**L'application produit-elle du WCS ?**

Oui, à partir de toute source raster.

**L'application produit-elle du WPS ?**

Oui, en installant l'extension dédiée. Cf <http://docs.geoserver.org/stable/en/user/extensions/wps/index.html>

**L'application produit-elle du WFS-T ?**

Oui, nativement, à partir de toute couche vectorielle.

**Fonctions transverses**

**Comment estimer la robustesse des composants ?**

La suite d'applications de l'IDS est actuellement utilisée en production avec plusieurs centaines d'utilisateurs.

**Quelle est l'homogénéité des environnements de développement / framework / langages ?**

Côté serveur, le langage majoritaire est JAVA, avec le framework MVC Spring. Il existe un module python permettant de faire de la recherche sur adresses. Côté client, le langage utilisé est exclusivement JAVASCRIPT.

**Comment estimez-vous, globalement, la facilité d'administration ?**

L'administration de la solution requiert sans aucun doute des compétences d'administration système.

**Comment estimez-vous la force de la communauté (réactivité en cas de demandes liées à des blocages, nombre de contributeurs, hétérogénéité des contributeurs) ?**

La communauté est rassemblée autour de listes de discussions actives et spécialisées (généraliste, développement, intégration continue). Cf <http://www.georchestra.org/community.html> pour plus d'informations.

**SGBD : quelle puissance en cas de requêtes complexes ?**

PostGreSQL et PostGIS sont des solutions matures et de niveau professionnel, qui sont en mesure de répondre à des requêtes spatiales complexes de manière performante.

**SGBD : quelle est l'intégration des fonctions géographiques de la plate-forme ? (les fonctions géographiques du SGBD sont-elles implémentées nativement ou bien le moteur cartographique utilisé déploie-t-il ses propres fonctions ?)**

Une réponse efficace à la directive INSPIRE

Le moteur cartographique utilise les fonctions spatiales de la librairie JTS, alors que PostGIS utilise celles de la librairie GEOS. GEOS est un port en C++ de JTS.

### **Quel est le niveau d'intégration des différents modules dans le portail, surtout en administration ?**

Il y a deux modalités d'accès à la suite d'applications spatiales de geOrchestra : soit directement en saisissant l'URL d'une application, soit à travers le CMS. Les différents modules et leurs consoles d'administration seront très prochainement intégrés au sein de l'IDS avec un bandeau permettant de commuter facilement de l'un à l'autre.

### **Prise en main / courbe d'apprentissage**

La prise en main requiert lecture de la documentation fournie. En cas de difficulté, la communauté répond rapidement (dans la journée, voire mieux) aux questions posées sur la liste de discussion georchestra-dev.

### **Sécurité : quelle authentification des utilisateurs ? (CAS, LDAP...)**

Les utilisateurs et les groupes sont stockés dans un arbre LDAP, servi par OpenLDAP. CAS est utilisé pour gérer l'authentification unique (SSO), tandis qu'un proxy de sécurité filtre l'accès aux applications en fonction de droits de l'utilisateur.

### **Sécurité : quelle sécurisation des échanges avec l'internaute sur les données sensibles ?**

La phase de login passe nécessairement en HTTPS. Il n'y a donc aucun risque d'interception des mots de passe dans l'échange client/serveur. Possibilité de configurer certaines applications en HTTPS (typiquement, GeoServer) s'il est nécessaire de sécuriser les flux OGC.

### **Sécurité : quelle réactivité de l'éditeur en cas de failles de sécurité et mises à jour ?**

Pour les briques CMS, catalogue et moteur cartographique, nous reposons sur la réactivité de l'éditeur, qui est très bonne. Pour les briques développées spécifiquement pour le projet, aucune faille de sécurité n'a été rapportée jusqu'à présent.

### **Quelle fréquence de mise à jour, quelle facilité de déploiement ?**

Une version stable tous les 6 mois, avec potentiellement des versions intermédiaires si le besoin est avéré. Le déploiement des applications se fait par upload des fichiers WAR des différents modules dans Apache Tomcat.

### **Liste des fonctionnalités**

La liste présentée ici est non exhaustive.

### **Visualiseur**

- se déplacer, zoomer dans la carte
- ajouter de nouvelles couches issue du catalogue ou d'un service WMS distant ;
- **modifier la sémiologie** d'une couche WMS locale ou distante et sauver le style ;
- **réaliser des requêtes sur une couche** pour filtrer les données affichées ;
- sauver la liste des couches ouvertes, la zone de visualisation dans un fichier de contexte ;

Une réponse efficace à la directive INSPIRE

- lire un fichier de contexte ;
- imprimer une carte au format PDF ;
- visualiser la carte dans une projection européenne (compatible INSPIRE) ;

### **Catalogue**

- profile français pour les métadonnées (compatible INSPIRE) ;
- moissonner des catalogues distants ;
- être moissonné par un catalogue distant ;
- gestion fine des droits de publication ;

### **Extracteur**

- choisir une zone d'extraction ;
- gestion des formats d'extraction (ECW, Geotiff, SHP, etc.) ;
- gestion des couches ;
- envoie d'un mail lorsque l'extraction est terminée.

### **Geopublisher**

- permettre de publier des données après téléchargement des données dans la fiche du catalogue.

### **Installation**



### **Guide d'installation**

Bien que l'objectif du projet soit de permettre la publication de binaires précompilés dans un dépôt central et de bénéficier d'un système qui modifie leur configuration pour une plate-forme cible, ce n'est pas la situation actuelle. Aujourd'hui, les configurations font encore partie du processus de construction, et il faut donc réaliser un build complet pour chaque plate-forme de déploiement.

### **Préconditions**

Les conditions suivantes doivent être satisfaites avant de déployer geOrchestra :

- Un annuaire LDAP est installé. Actuellement, le module intégré d'annuaire LDAP n'est pas déployé, un annuaire LDAP externe est donc requis. Nous employons habituellement OpenLDAP.

## Une réponse efficace à la directive INSPIRE

- Un certificat SSL pour l'adresse publique du serveur. Pour utiliser le déploiement standard, il faut disposer d'une installation apache2 avec https et un certificat configuré.
- Une base de données pour permettre à Geonetwork de stocker ses données. La connexion à la base est configurée dans la configuration Geonetwork (voir plus loin)

En plus des utilisateurs, l'annuaire LDAP doit contenir la liste des groupes/rôles, et évidemment un utilisateur administrateur. Chaque utilisateur DOIT être décrit avec certains champs incluant:

- mail
- uid (à confirmer)
- cn (à confirmer)

L'annuaire doit également respecter certaines règles pour le nommage des groupes/rôles.

- Les groupes/rôles préfixés par EL\_ seront appliqués aux groupes Geonetwork
- Les groupes/rôles préfixés SV\_ sont des rôles communs à tous les modules (dans Geonetork, les rôles SV\_ correspondent aux profils, dans d'autres applications ils ont une signification différente)
  - SV\_ADMIN donne des permissions administrateur dans toutes les applications (hors geoserver)
  - SV\_EDITOR donne des droits en écriture dans une application si cette notion a un sens (actuellement, cela concerne seulement Geonetwork et Mapfishapp)
  - SV\_REVIEWER donne des droits de relecteur validateur (actuellement, cela concerne seulement Geonetwork)
  - SV\_USER donne seulement des droits en lecture seule dans toutes les applications mais, l'utilisateur étant authentifié, il peut bénéficier d'autres permissions (dépendantes de l'application)
  - GS\_ADMIN est un rôle spécial pour l'administration de Geoserver (attention : ADMINISTRATOR pour GS2). SV\_ADMIN donne un accès complet à la configuration de geoserver, mais GS\_ADMIN donne accès limité à la configuration des permissions d'accès aux couches. Un utilisateur avec un rôle GS\_ADMIN ""doit"" également avoir un rôle GS\_ADMIN\_<FOO>. Le <FOO> indique quel "regroupement de couches" l'utilisateur/administrateur est en droit d'administrer.

## **Configuration**

Tous les projets requièrent une configuration préalable pour personnaliser le projet pour une plate-forme particulière. En général, les configurations sont stockées dans le répertoire `<projet>/src/<platform_id>`. On y trouve habituellement un fichier de propriétés pour filtre maven qui définit les principaux paramètres de configuration (il y en a davantage dans le répertoire `src/main/webapp/WEB-INF`, mais il n'est pas nécessaire de les modifier pour un déploiement simple, et la structure de ces fichiers sont très différents pour chaque projet). Pour chaque projet, il est recommandé de copier les paramètres de configuration d'une plate-forme de déploiement existante pour ensuite les adapter à la nouvelle plate-forme.

Après avoir édité les fichiers de configuration de chaque projet, une section *profile* doit être ajoutée dans le `pom.xml` racine comme suit (notez que `platform_id` doit correspondre à l'identifiant que vous avez choisi pour désigner la plate-forme cible.) :

Une réponse efficace à la directive INSPIRE

```
<profile>
    <id>platform_id</id>
    <properties>
        <server>platform_id</server>
    </properties>
</profile>
```

Une fois tous les fichiers de configuration mis à jour, on peut construire tous les projets.

### **Construction**

Depuis la racine des sources, exécutez maven en spécifiant la plate-forme de déploiement cible et la tâche (typiquement install)

```
mvn install -P<configurationkey>
```

Lorsqu'exécuté dans le répertoire racine, tous les projets seront construits. Lorsqu'exécuté dans un module (comme extractorapp), seul ce module sera construit.

Ceci prendra beaucoup de temps. Lorsque tous les projets sont construits, on trouvera dans le dépôt maven local un fichier war pour chaque module, avec platform\_id ajouté en classifier. Par exemple, si vous construisez `mvn_install -Pdev` dans le répertoire cas-server-webapp, l'artefact `cas server-webapp-1.0-dev.war` sera placé dans le dépôt local maven. Dans mon cas, le fichier se trouve dans `/home/username/.m2/repository/org/georchestra/cas-server-webapp/1.0/cas-server-webapp-1.0-dev.war`

Une fois tous les artefacts construits, ils peuvent être déployés avec le module server-deploy.

### **Déploiement**

La première étape à effectuer est la création d'un script de déploiement. Le nom de ce script est important, il doit suivre la syntaxe `<platform_id>DeployScript.groovy`. Voir la section technique ci-dessous pour davantage d'informations sur la façon de rédiger un script de déploiement.

La deuxième étape est d'ajouter les informations de connexion pour le serveur de destination dans le fichier de configuration de maven. En règle générale, ce fichier est situé dans `$HOME/.m2/settings.xml`. Ce fichier n'étant pas spécifique au processus de déploiement de geOrchestra, les informations sur sa syntaxe se trouvent sur le site officiel, à l'adresse suivante : <http://maven.apache.org/settings.html#Servers>. Tous les paramètres ne sont pas utiles pour un déploiement, seulement :

- identifiant (id, celui-ci doit correspondre au profil maven ajouté dans le pom.xml)
- l'hôte (host, cela n'est pas dans la documentation de maven, mais correspond au nom d'hôte du serveur cible)
- nom d'utilisateur (username)
- mot de passe (password, optionnel)
- clé privée (privateKey, optionnel)
- Phrase de passe (passphrase, optionnel)

Une fois le script écrit, les projets peuvent être déployés en exécutant :

## Une réponse efficace à la directive INSPIRE

- mvn -Pfull,platform\_id -- Ceci va déployer tous les fichiers war et configurer tous les systèmes tiers tels que openLDAP, les certificats serveur, la configuration apache, la configuration tomcat, etc...
- mvn -Pupdate,platform\_id -- Ceci va déployer tous les fichiers war, mais ne touche pas au reste du système
- mvn -P<app>,platform\_id -- Substituez <app> pour l'application que vous souhaitez déployer. Par exemple: mvn -Pcas,platform\_id

### **Informations techniques**

#### **Mécanisme de déploiement**

**Le mécanisme de déploiement consiste en deux modules :**

- server-deploy
- server-deploy-support

Le module server-deploy-support module contient des classes Java et Groovy (qui sont indépendantes de la plateforme) pour simplifier l'écriture de scripts de déploiement vers tous systèmes et serveurs web. Des exemples de classes :

- SSH - fournit des commandes scp et ssh indépendantes de la plateforme, permettant de déployer vers tout serveur équipé d'ssh depuis windows ou linux ;
- SSHTomcatDeployer - Permet le dépôt de fichiers war sur un serveur tomcat en trois lignes de code. Cela inclut la copie du fichier sur le serveur distant, la mise à jour des wars existants, le redémarrage de tomcat si nécessaire.

Le module server-deploy contient les scripts actuels pour réaliser les déploiements. Il y a des profils pour effectuer un déploiement complet, pour mettre à jour un unique module, ou tous les modules. Les scripts sont très simples à écrire, par exemple:

```
def ssh = new SSH(log:log,settings:settings,host:"c2cpc83.campocamp.com")  
  
def deploy = new C2CDeploy(project,ssh)  
deploy.deploy()
```

Le code ci-dessus déploie en utilisant la configuration C2CDeploy par défaut, qui consiste en deux serveurs tomcat. Ceci n'est bien sûr pas applicable à toutes les situations, l'exemple suivant montre comment déployer Geoserver sur un serveur et toutes les autres applications sur un autre serveur.

```
def artifacts = new Artifacts(project, Artifacts.standardGeorchestraAliasFunction)  
def ssh = new SSH(log:log,settings:settings,host:"server1")  
def server1Deployer = new SSHWarDeployer(  
    log: log,  
    ssh: ssh,  
    projectProperties: projectProperties,  
    webappDir: "/srv/tomcat/tomcat1/webapps",  
    startServerCommand: "sudo /etc/init.d/tomcat-tomcat1 start",  
    stopServerCommand: "sudo /etc/init.d/tomcat-tomcat1 stop"  
)  
server1Deployer.deploy(artifacts.findAll{!it.name.contains("geoserver")})
```

## Une réponse efficace à la directive INSPIRE

```
def geoserverArtifact = artifacts.find{it.name.contains("geoserver")}  
if (geoserverArtifact != null) {  
    def geoserverSSH = ssh.changeHost("server2")  
    def geoserverDeployer = tomcat1Deployer.copy(ssh: geoserverSSH)  
    geoserverDeployer.deploy()  
}
```

Le code ci-dessus est dans `trunk/server-deploy/exampleDeployScript.groovy` et est abondamment commenté pour expliquer le propos de chaque ligne.

Pour résumer. Le module server-deploy fournit un moyen d'écrire facilement des scripts de déploiement pour déployer un système entier sur un ou plusieurs serveurs. Les objectifs de ce module sont :

- Fournir un moyen décrire très facilement des scripts de déploiement.
- Etre indépendant du système, de façon à ce que qu'un script fonctionne sur toute plate-forme.
- Ne pas nécessiter d'installation autre que maven et java.
- Rester très flexible, pour qu'il soit facile d'écrire des scripts qui déploie tous les war sur un seul serveur, ou un module sur plusieurs serveurs pour des questions de charge et tous les autres sur un autre serveur.

Actuellement, le module server-deploy-support fournit un ensemble basique de modes de déploiement, mais il peut être étendu avec d'autres classes pour faciliter l'écriture de scripts de déploiement vers d'autres types d'environnement.

## **Informations techniques diverses**

### **Java SSL, Keystores and Truststores**

Un keystore stocke les certificats d'un serveur et les secrets associés, et est utilisé quand un serveur veut s'authentifier auprès d'un autre serveur. Si vous voulez qu'un serveur tomcat (par exemple) bénéficie d'un certificat, vous devrez créer un keystore et y déposer le certificat. Les certificats sont souvent au format PEM, vous devez alors utiliser un script comme: <https://github.com/jesseeechar/jvm-security-scripts/blob/master/ImportDem.java> ou <https://github.com/jesseeechar/jvm-security-scripts/blob/master/ImportDem.scala> pour convertir le certificat PEM et l'installer dans le keystore. Naturellement, vous avez besoin d'un keystore avant d'installer quoi que ce soit dedans; vous pouvez, en créer un en utilisant le script [https://github.com/jesseeechar/jvm-security-scripts/blob/master/create\\_empty\\_Keystore](https://github.com/jesseeechar/jvm-security-scripts/blob/master/create_empty_Keystore) qui crée un keystore vierge..

Pour que deux serveurs dialoguent de façon sécurisée, l'un doit présenter un certificat et l'autre doit faire confiance à ce certificat. C'est ici que le truststore intervient. Par défaut, les JVM sont fournies avec un truststore approuvant les principaux fournisseurs de certificats. Si vous avez acquis un certificat auprès de ces fournisseurs, tout va bien. Sinon, vous devez créer un keystore (voir les scripts ci-dessus), puis importer le certificat serveur dans le truststore en utilisant l'un de ces scripts : <https://github.com/jesseeechar/jvm-security-scripts/blob/master/InstallCert.java> ou <https://github.com/jesseeechar/jvm-security-scripts/blob/master/InstallCert.scala>. Ces scripts demandent au serveur cible leur certificat, puis l'installent dans le truststore.

Un point majeur est que le certificat est lié au nom d'hôte. Si le serveur a plusieurs alias, vous devez choisir lequel utiliser.

Une réponse efficace à la directive INSPIRE

### **Procédure post-installation**

#### **Configuration des référentiels**

mapfishapp et extractorapp proposent chacun une liste déroulante de couches référentes. Pour que ces couches apparaissent, il faut configurer correctement GeoServer.

Tout d'abord il faut générer une couche avec une seule colonne attributaire, généralement le nom de l'entité. Par exemple pour une couche *commune*, il faut un champ attributaire *nom\_commune*. Le nom du champ importe peu.

Ensuite il faut importer cette couche dans GeoServer dans un namespace spécifique appelé *geob\_loc* et l'activer dans le service WFS.

La couche devrait apparaître dans la liste déroulante.

#### **Mapfishapp**

#### **Personnalisation de la liste des WMS**

La liste des services WMS proposés dans la fenêtre de chargement des couches peut être personnalisée à l'aide du fichier de configuration *src/main/webapp/app/js/GEOB\_wmslist.js*

#### **Personnalisation des couches chargées au démarrage**

Cela s'effectue dans le fichier à la racine de la mapfishapp : *src/main/webapp/dev-default.wmc*

#### **Précédure d'installation du module search\_address**

L'application search-address est basée sur le framework web Pylons [1], et utilise zc.buildout (a.k.a. buildout) pour son installation et sa configuration.

Cette page décrit comment installer et configurer cette application.

L'installation de la base de données adresse n'est pas décrite ici. Nous supposerons ici que la base de données est fonctionnelle.

#### **Dépendances**

Vous devez installer certains paquets :

```
apt-get install python2.6 python2.6-dev lippq-dev libgeos-cl libgeos-dev  
apt-get install package build-essentials
```

vous devez installer PostgreSQL >= 8.4, car ce module utilise le wildcard \* de FTS, disponible seulement à partir de cette version.

#### **Installation**

Il vous faut d'abord un environnement buildout en utilisant cette commande :

```
python bootstrap.py --version 1.5.2 --distribute --download-base http://pypi.campocamp
```

Créez un fichier de configuration de buildout, pour cela copier le fichier *buildout\_sample.cfg* dans un nouveau fichier. Vous pouvez choisir n'importe quel nom pour ce fichier, nous supposerons que le nom du fichier sera *buildout\_mine.cfg* dans ce qui suit.

Éditez le fichier *buildout\_mine.cfg* et définissez les variables comme appropriées. Les noms de variables et les commentaires devraient vous guider pour comprendre ce que font ces

Une réponse efficace à la directive INSPIRE

variables.

Lancez l'installation du buildout et configurez search-address et ses dépendances :

```
./buildout/bin/buildout -c buildout_mine.cfg
```

Le processus de buildout devrait se terminer avec l'affichage de ce qui suit :

```
***** PICKED VERSIONS *****
[versions]
***** /PICKED VERSIONS *****
```

L'installation est alors terminée. Vous pouvez tester que le projet fonctionne correctement en lançant l'application search-address dans le server web inclus :

```
./buildout/bin/paster serve searchaddress_dev.ini
```

Et doit renvoyer ceci :

```
Starting server in PID 1358.
serving on 0.0.0.0:5000 view at http://127.0.0.1:5000
```

Vous pouvez tester que le service web addresses fonctionne en ouvrant l'URL suivante dans votre navigateur :

```
http://<hostname>:5000/addresses?limit=20&lang=fr&attrs=street%2Chousenumber%2Ccity&que
```

Entrez CTRL+C pour arrêter le server web.

### **Exécution dans Apache**

Pour exécuter l'application search-address dans Apache vous devez avoir le module mod\_wsgi d'Apache installé et activé. Si vous utilisez une distribution Debian-like vous pouvez installer simplement le paquet Debian libapache2-mod-wsgi.

L'application est fournie avec une configuration Apache. Pour exécuter l'application dans Apache vous devez inclure cette configuration dans votre configuration globale d'Apache. Cela est réalisée en utilisant la directive Include d'Apache :

```
Include /path/to/search-address/apache/wsgi.conf
```

Vous pouvez maintenant redémarrer ou recharger la configuration d'Apache et tester l'URL suivante pour vous assurer que la recherche d'adresse est fonctionnelle.

```
http://<hostname>/addrapp/addresses?limit=20&lang=fr&attrs=street%2Chousenumber%2Ccity&que
```

### **Gestion des droits et sécurité**

#### **Introduction**

La mise en place de droit d'accès est fondamentale au sein d'une Infrastructure de Données Spatiales. Cette page a pour objectif d'expliquer son fonctionnement au sein de geOrchestra.

## Une réponse efficace à la directive INSPIRE

La première partie décrit les interfaces de gestion des accès en fonction de l'outil utilisé. La section suivante illustre le concept de groupe et de région au sein de geOrchestra.

Vous pouvez accéder à la page [Guide d'installation](#) pour d'autres informations à ce sujet.

### Gestion

#### Catalogue

##### Principes

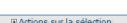
La gestion de droits du catalogue permet d'attribuer à des groupes d'utilisateurs des permissions sur les métadonnées. Dans le catalogue, on appelle ces permissions des priviléges.

- le privilège Publier autorise la consultation de la fiche par l'interface ou par le protocole CSW . Associé au groupe spécial Tous, ce privilège autorise tout public à consulter la métadonnée.
- le privilège Télécharger autorise le téléchargement des données. Une section Options de transfert apparaît dans la fiche avec un lien de téléchargement.
- le privilège Carte interactive autorise le chargement des données sur la carte interactive du catalogue. Une section Options de transfert apparaît dans la fiche avec un lien Carte interactive.

Pour résoudre les priviléges d'un utilisateur, les règles suivantes sont appliquées :

- les administrateurs du catalogue ont bien évidemment tous les priviléges,
- un auteur reçoit automatiquement tous les priviléges sur les fiches qu'il a produites,
- un utilisateur reçoit les priviléges attribués à ses groupes.

#### Interface

Les priviléges peuvent être appliqués à une unique fiche avec le menu  ou, après recherche de fiches, à un ensemble de fiches avec le menu .

L'interface de gestion des priviléges se présente sous la forme d'une matrice avec les groupes en ligne et les priviléges en colonne. En cochant une case, on attribue à un groupe le privilège lui permettant d'effectuer une action. On peut cocher autant de cases que souhaité.



#### Serveur cartographique

- via gssec pour GeoServer 1 :

- **avantage :**

- permet de structurer les ACL en "groupes" pour ensuite en déléguer la gestion.
    - Ne nécessite pas un compte administrateur geoserver.
    - Requiert un accès système pour gérer les délégations.

- **inconvénient :**

- ergonomie non adaptée à un grand nombre de couches ou de groupes.

## Une réponse efficace à la directive INSPIRE

- Stabilité à éprouver.
- Pas de nettoyage automatique pour les groupes/couches supprimés.
- GeoServer 2 et supérieur :
  - **avantage :**
    - intégré, stable,
    - fonction de recherche adaptée à un grand nombre de couches,
    - pas d'accès système nécessaire.
  - **inconvénient :**
    - ne connaît pas la liste des groupes LDAP qui doivent être entrés comme des mots clefs, pas de délégation possible.
    - Pas de nettoyage automatique pour les groupes/couches supprimés.

### **Notion de groupe**

On peut comprendre les groupes comme des "tags" que l'on applique à un utilisateur.

- EL\_\* : groupe structurel (accès en fonction de la structure)
- SV\_\* : groupe fonctionnel (accès en fonction des fonctionnalités, si j'ose dire)

#### **Note**

Par défaut, toutes les couches peuvent être accéder en lecture, aucune ne l'est en écriture.

Exemple de groupe structurel :

- EL\_C2C
- EL\_adherent
- EL\_nonadherent
- EL\_OSM
- EL\_BRETAGNE
- EL\_BREST

*user1* appartient à EL\_C2C et EL\_adherent. Par contre *user2* appartiendra à EL\_C2C, EL\_adherent et EL\_OSM.

Exemple de groupe fonctionnel :

- SV\_ADMIN : big boss
- SV\_USER : peut s'authentifier
- SV\_EDITOR : édition du catalogue
- SV\_REVIEWOR : si l'organisme a des sous, des utilisateurs qui peuvent relire les fiches de catalogues

### **Notion de région**

## Une réponse efficace à la directive INSPIRE

Il s'agit plutôt de groupe d'ACL plutôt que de "région" au sens localisation. L'intérêt est de pouvoir permettre la délégation de la gestion des ACL des couches à un utilisateur.

### Exemple :

L'utilisateur *user2* appartient au groupe *EL\_ADMIN\_OSM*, on crée un fichier de configuration *OSM.properties* qui fait le lien entre le nom du groupe d'ACL et le groupe LDAP (*EL\_ADMIN\_OSM* par exemple) et cet utilisateur, après qu'on lui ait donné un droit sur une couche, pourra gérer les accès pour ses collègues. Il ne verra et pourra modifier que les acl au sein de ce groupe/région.

Une version anglaise (*Installation Guide*) et en espagnol (*Guía de instalación*) sont disponibles.

## **Guía de instalación**

Aunque el objetivo del proyecto sea de permitir la publicación de binarios precompilados en un repositorio central y de beneficiar de un sistema que modifica su configuración para una plataforma dada, no es la situación actual. Hoy día, las configuraciones siguen siendo parte del proceso de construcción, y todavía hay que realizar un build completo para cada plataforma de desarrollo.

### **Precondiciones**

Las condiciones siguientes tienen que ser satisfechas antes de desplegar geOrchestra:

- Un directorio LDAP esta instalado. Actualmente, el modulo integrado de directorio LDAP no esta desplegado, entonces un directorio LDAP externo esta requerido. Utilizamos generalmente OpenLDAP.
- Un certificado SSL para la dirección pública del servidor. Para utilizar un depliegue standar, la instalación apache2 tiene que contar con https y un certificado configurado.
- Una base de datos para permitir a Geonetwork de almacenar sus datos. La conexión a la base esta configurada en la configuración de GeoNetwork (ver más abajo)

Además de los usuarios, el anuario LDAP debe contener la lista de grupos/roles, y obviamente un usuario administrador. Cada usuario TIENE QUE estar descrito con ciertos campos, incluyendo:

- correo
- uid (queda para confirmar)
- cn (queda para confirmar)

El directorio tiene que respetar también ciertas reglas para los nombres de grupos/roles.

- Los grupos/roles con el prefijo *EL\_* serán aplicados a los grupos Geonetwork
- Los grupos/roles con el prefijo *SV\_* son roles comunes a todos los modulos (en Geonetwork, los roles *SV\_* corresponden a los perfiles, en otras aplicaciones tienen otro significado)
  - *SV\_ADMIN* da permisos de administrador en todas las aplicaciones (menos geoserver)
  - *SV\_EDITOR* da derechos de escritura en una aplicación si está noción tiene un sentido (actualmente, solo para Geonetwork y Mapfishapp)
  - *SV\_REVIEWER* da derechos de relector validator (actualmente, solo para Geonetwork)

## Une réponse efficace à la directive INSPIRE

- SV\_USER da solamente derechos de lectura sola en todas las aplicaciones pero el usuario, estando autenticado, puede beneficiar de otros permisos (dependiendo de la aplicación)
- GS\_ADMIN es un rol especial para la administración de Geoserver (Ojo: ADMINISTRADOR para GS2). SV\_ADMIN da un acceso completo a la configuración de geoserver, pero GS\_ADMIN da acceso limitado a la configuración de los permisos de acceso a las capas. Un usuario con un rol GS\_ADMIN "debe" tener también un rol GS\_ADMIN\_<FOO>. El <FOO> indica cual "agrupación de capas" el usuario/administrados tiene el derecho de administrar.

## Configuración

Todos los proyectos necesitan una configuración previa para adaptar el proyecto a una plataforma en particular. En general, las configuraciones están almacenadas en la carpeta <proyecto>/src/<plataforma\_id>. En esta carpeta se encuentran normalmente un archivo de propiedades para filtro maven que define los principales parámetros de configuración (hay más parámetros en la carpeta src/main/webapp/WEB-INF, pero no es necesario modificarlos para un despliegue simple, y la estructura de estos archivos es muy diferente según los proyectos). Para cada proyecto, recomendamos de copiar en un primer tiempo los parámetros de configuración de una plataforma de despliegue existente, para adaptarlos en un segundo tiempo a la nueva plataforma.

Luego de haber editado los archivos de configuración de cada proyecto, se tiene que añadir una sección *profile* en el pom.xml raíz de la manera siguiente (Ojo: platform\_id tiene que corresponder al identificador que eligiste para designar la nueva plataforma.)

```
<profile>
    <id>platform_id</id>
    <properties>
        <server>platform_id</server>
    </properties>
</profile>
```

Una vez actualizados todos los archivos de configuración, podemos construir todos los proyectos.

## Construcción

Desde la raíz de las fuentes, ejecuta maven especificando la plataforma de despliegue y la tarea (en general: install)

```
mvn install -P<configurationkey>
```

Cuando este comando está ejecutado en la carpeta raíz, todos los proyectos serán construidos. Cuando está ejecutado en un módulo (extractorapp por ejemplo), solo este módulo será construido.

La construcción puede tomar mucho tiempo. Una vez construidos todos los proyectos, puedes encontrar en el repositorio maven local un archivo war para cada módulo, con platform\_id añadido en classifier. Por ejemplo, si construyes mvn install -Pdev en la carpeta cas-server-webapp, el archivo (llamado artefacto) cas-server-webapp-1.0-dev.war sera creado en el repositorio local maven. En mi caso, el archivo se encuentra en /home/username/.m2/repository/org/georchestra/cas-server-webapp/1.0/cas-server-webapp-1.0-dev.war

## Une réponse efficace à la directive INSPIRE

Una vez todos los artefactos construidos, pueden ser despliegados con el módulo server-deploy.

### **Despliegue**

La primera etapa es la creación de un script de despliegue. El nombre del script es importante, tiene que seguir la sintaxis <platform\_id>DeployScript.groovy. Ver la sección técnica más abajo para más información sobre la manera de redactar un script de despliegue.

La segunda etapa es de añadir las informaciones de conexión para el servidor de destinación en el archivo de configuración de maven. En general, este archivo se ubica en \$HOME/.m2/settings.xml. Este archivo no es específico al proceso de despliegue de geOrchestra, y las informaciones sobre su sintaxis están en el sitio oficial, a la dirección siguiente: <http://maven.apache.org/settings.html#Servers>. No todos los parámetros son útiles para un despliegue, solamente:

- identificados (id, tiene que corresponder al perfil maven añadido en el pom.xml)
- el host (host, no está en la documentación de maven, pero corresponde al nombre de host del servidor de despliegue)
- nombre de usuario (username)
- contraseña (password, opcional)
- clave privada (privatekey, opcional)
- passphrase (passphrase, opcional)

Una vez escrito el script, los proyectos pueden ser desplegados ejecutando:

- mvn -Pfull,platform\_id -- Eso desplegará todos los archivos war y configurará todos los sistemas terceros como openLDAP, los certificados servidor, la configuración apache, la configuración tomcat, etc.
- mvn -Pupdate,platform\_id -- Eso desplegará todos los archivos war, pero no tocará al resto del sistema
- mvn -P<app>,platform\_id -- Reemplaza <app> con la aplicación que quieras desplegar. Por ejemplo: mvn -Pcas,platform\_id

## **Informaciones técnicas**

### **Mecanismo de despliegue**

**El mecanismo de despliegue consiste en dos módulos:**

- server-deploy
- server-deploy-support

El módulo server-deploy-support contiene clases Java y Groovy (que son independientes de la plataforma) para simplificar la escritura de los scripts de despliegue hacia todo tipo de sistemas y servidores web. Unos ejemplos de clases:

- SSH - provee comandos scp y ssh independientes de la plataforma, permite desplegar hacia cualquier servidor con SSH desde windows o linux
- SSHTomcatDeployer - permite depositar archivos war sobre un servidor tomcat en tres líneas de código. Incluye la copia del archivo sobre el servidor distante, la actualización de los war existentes y la reinicialización de tomcat si es necesaria.

Une réponse efficace à la directive INSPIRE

El módulo server-deploy contiene los scripts para realizar los despliegues. Existen perfiles para hacer un despliegue completo, para actualizar un único módulo, o todos los módulos. Los scripts son muy simples, por ejemplo:

```
def ssh = new SSH(log:log,settings:settings,host:"c2cpc83.campocamp.com")  
  
def deploy = new C2CDeploy(project,ssh)  
deploy.deploy()
```

Este código despliega utilizando la configuración C2CDeploy por omisión, lo que consiste en dos servidores tomcat. Obviamente no es aplicable a todas las situaciones, el ejemplo siguiente muestra como desplegar Geoserver sobre un servidor y todas las otras aplicaciones sobre un otro servidor.

```
def artifacts = new Artifacts(project, Artifacts.standardGeorchestraAliasFunction)  
def ssh = new SSH(log:log,settings:settings,host:"server1")  
def server1Deployer = new SSHWarDeployer(  
    log: log,  
    ssh: ssh,  
    projectProperties: projectProperties,  
    webappDir: "/srv/tomcat/tomcat1/webapps",  
    startServerCommand: "sudo /etc/init.d/tomcat-tomcat1 start",  
    stopServerCommand: "sudo /etc/init.d/tomcat-tomcat1 stop"  
)  
server1Deployer.deploy(artifacts.findAll{!it.name.contains("geoserver")})  
  
def geoserverArtifact = artifacts.find{it.name.contains("geoserver")}  
if (geoserverArtifact != null) {  
    def geoserverSSH = ssh.changeHost("server2")  
    def geoserverDeployer = tomcat1Deployer.copy(ssh: geoserverSSH)  
    geoserverDeployer.deploy()  
}
```

Este código está en `trunk/server-deploy/exampleDeployScript.groovy` y tiene muchos comentarios para explicar cada línea.

Para resumir. El módulo server-deploy provee una forma de escribir fácilmente scripts de despliegue para desplegar un sistema entero sobre uno o varios servidores. Los objetivos de este módulo son:

- Proveer una manera de describir muy fácilmente scripts de despliegue.
- Ser independiente del sistema, de tal manera que un script funcione sobre cualquier plataforma.
- No necesitar ninguna otra instalación que maven y java.
- Quedar muy flexible, para que sea fácil escribir scripts que desplieguen todos los war en un solo servidor, o un módulo en varios servidores para repartir la carga, y todos los otros módulos sobre un otro servidor.

Actualmente, el módulo server-deploy-support provee una lista básica de modos de despliegue, pero puede ser extendido con otras clases para facilitar la escritura de scripts de despliegue hacia otros tipos de entorno.

### **Algunas informaciones técnicas**

### **Java SSL, Keystores and Truststores**

Un keystore almacena los certificados de un servidor y los secretos asociados, y esta utilizado cuando un servidor se quiere autenticar en un otro servidor. Si quieres que un servidor tomcat (por ejemplo) utilice un certificado, tienes que crear un keystore y depositar el certificado adentro. Los certificados son generalmente en el formato PEM, en este caso tienes que utilizar un script como:  
<https://github.com/jesseeechar/jvm-security-scripts/blob/master/ImportDem.java> o  
<https://github.com/jesseeechar/jvm-security-scripts/blob/master/ImportDem.scala> para convertir el certificado PEM e instalarlo en el keystore. Naturalmente, necesitas un keystore antes de poder instalar certificados adentro ; puedes crear uno con el script [https://github.com/jesseeechar/jvm-security-scripts/blob/master/create\\_empty\\_Keystore](https://github.com/jesseeechar/jvm-security-scripts/blob/master/create_empty_Keystore) que crea un keystore vacío.

Para que dos servidores dialoguen de manera segura, uno tiene que presentar un certificado y el otro tiene que confiar en este certificado. Aquí interviene el truststore. Por omisión, las JVM vienen con un truststore que contiene los principales proveedores de certificados. Si compraste un certificado a uno de estos proveedores, no hay nada más que hacer. Sino, tienes que crear un keystore (ver los scripts más arriba), luego importar el certificado servidor en el truststore con uno de estos scripts:  
<https://github.com/jesseeechar/jvm-security-scripts/blob/master/InstallCert.java> o  
<https://github.com/jesseeechar/jvm-security-scripts/blob/master/InstallCert.scala>. Estos scripts piden el certificado al servidor distante, y luego lo instalan en el truststore.

Un punto importante es que el certificado está ligado con el hostname. Si el servidor tiene varios aliases, tienes que elegir cual utilizar.

### **Installation Guide**

While the goal of the project is to be able to publish raw built artifacts to a central repository and have a configuration system that modifies their configuration for a particular deployment platform, that is not currently the situation. As it stands the configurations are built along with the artifacts and as such one must do a full build for each platform you want to deploy to.

### **Preconditions**

The following precondition must be met before deploying georchestra

- An LDAP server is installed, currently the Embedded-LDAP module is not deployed so another LDAP server is required. Typically we use OpenLDAP.
- A certificate for the outward facing server. To use the default C2CDeploy an apache2 install is required with a certificate configured as well as https.
- A database must be available for Geonetwork to store its data in. The connection parameters for the database is configured as part of the geonetwork configuration (See below)

The LDAP needs to be configured with users and Groups/Roles (and obviously an admin user). Each user MUST have certain fields including:

- mail
- uid (maybe)
- cn (maybe)

There are also special rules for Groups/Roles in the LDAP.

## Une réponse efficace à la directive INSPIRE

- Groups/Roles starting with EL\_ will be mapped to Geonetwork groups
- Groups/Roles starting with SV\_ are roles that are common for all applications. (In the case of Geonetwork SV\_ roles map to profiles, in other applications it has different meaning)
  - SV\_ADMIN gives administrative rights to all applications
  - SV\_EDITOR gives editing rights within an application if it applies (Currently this is observed by Geonetwork and Mapfishapp only)
  - SV\_Reviewer gives review rights (Currently only observed by Geonetwork)
  - SV\_User give read only rights in all applications but since the user is logged in it may give extra access to the application (application dependent)
- GS\_ADMIN is a special role for administrating Geoserver. SV\_ADMIN give full admin access to geoserver but GS\_ADMIN gives access to the geoserver security application for configuring access to each layer in geoserver. A user with GS\_ADMIN "must" also have a GS\_ADMIN\_<FOO> role. the <FOO> indicates what "region/group" that administrator has rights to administer.

### **Configuration**

Each of the projects require a configuration to customize that project for a particular deployment platform. In general the configurations are stored in: <project>/src/<platform\_id>. Normally there is a maven.filters file which defines the primary configuration parameters (there are more in the src/main/webapp/WEB-INF folder as well but normally they do not need to be modified and are different for each project). It is recommended for each project to copy the configuraiton parameters of another deployment platform and modify it for the new platform. For example update the host files.

After adding the configuration files for each project a *profile* section must be added to the root pom.xml as follows (note that platform\_id needs to correspond to the identifier you chose to identify the target deploy platform.):

```
<profile>
    <id>platform_id</id>
    <properties>
        <server>platform_id</server>
    </properties>
</profile>
```

Once all configuration files have been updated one can build all projects.

### **Building all modules**

From the root directory of the checkout execute maven specifying the server you want to build for and the task (typically install)

```
./mvn install -P<configurationkey>
```

When executed in the root directory all project will be build. When the command is executed in a submodule (like extractorapp) only that module will be built.

This will take a very long time. After all the projects are built each webapp will have a war file published to your local maven repository with the platform\_id appended as a classifier. For example if you build \* mvn install -Pdev\* in the cas-server-webapp directory, the artifact

## Une réponse efficace à la directive INSPIRE

\* cas-server-webapp-1.0-dev.war\* will be put in your local maven repository. In my case the file is in /home/username/.m2/repository/org/georchestra/cas-server-webapp/1.0/cas-server-webapp-1.0-dev.war  
Once all of the artifacts are built then they can be deployed with the server-deploy module.

### **Building submodules**

By default, the mvn command uses the "all" profile. This profile builds all submodules for the specified configuration. You may want to build only a subset of modules. To do this, deactivate the all modules and specify a module profile list

```
./mvn install -P-all,extractorapp,<configurationkey>
```

Notice the "-all" that deactivates the whole project building, and the "extractorapp" profile enabling the extractor application build.

You may use the following profiles :

- all - builds all modules (default option)
- -all - deactivates the whole project build
- cas-web-server - builds the CAS server
- geonetwork - builds the geonetwork application
- doc - builds the documentation application
- extractorapp - builds the extractor application
- mapfishapp - builds the viewer application
- catalogapp - builds the catalog light client application
- security-proxy - builds the security proxy application

Other submodules are dependencies and cannot be deactivated.

### **Deploy**

The first step that needs to be done is to create a deployScript. The name of the file is important, it must follow the form <platform\_id>DeployScript.groovy. See the technical information section below for more detailed information on how to write a deployScript.

The second step is to add the credentials for the target server in the maven settings file. Typically that file is in \$HOME/.m2/settings.xml. The normal maven settings is used, so read <http://maven.apache.org/settings.html#Servers> for details. Not all parameters are used by deploy. Only

- id
- host (not shown in the maven page, it is the hostname of the target server)
- username
- password (optional)
- privateKey (optional)
- passphrase (optional)

Now the system can be deployed by executing:

## Une réponse efficace à la directive INSPIRE

- mvn -Pfull,platform\_id -- This will deploy all war files as well as configure all ancillary systems that are required like openLDAP, server certificates, apache configuration, tomcat configuration, etc...
- mvn -Pupdate,platform\_id -- This will deploy all war files but leave the rest of the system alone
- mvn -P<app>,platform\_id -- Substitue <project> for the app you want to deploy. For example: mvn -Pcas,platform\_id

### **Technical Information**

#### **Deployment Mechanism**

##### **The deploy mechanism consists of 2 modules:**

- server-deploy
- server-deploy-support

The server-deploy-support module contains Java and Groovy classes (which are platform independent) to make it simple to write deploy scripts for deploying to any system of servers and web containers. Examples of classes are:

- SSH - provide scp and ssh commands in a platform independent manner so one can deploy to an ssh compliant server from windows or linux.
- SSHTomcatDeployer - Allows delivering one or more war files to a tomcat with around 3 lines of code. It includes copying the files to the remote server, updating the existing wars and restarting the tomcat if necessary.

The server-deploy module contains the actual scripts for performing the deployment. It has profiles for performing a full deploy, and upgrade of a single module or a upgrade of all modules. The scripts are very simple to write for example:

```
def ssh = new SSH(log:log,settings:settings,host:"c2cpc83.campocamp.com")

def deploy = new C2CDeploy(project,ssh)
deploy.deploy()
```

The above code deployed using the default C2CDeploy configuration which consists of 2 tomcat servers. Naturally that is not applicable to all situations so the following is an example of deploying Geoserver to one server and all other applications to another server.

```
def artifacts = new Artifacts(project, Artifacts.standardGeorchestraAliasFunction)
def ssh = new SSH(log:log,settings:settings,host:"server1")
def server1Deployer = new SSHWarDeployer(
    log: log,
    ssh: ssh,
    projectProperties: projectProperties,
    webappDir: "/srv/tomcat/tomcat1/webapps",
    startServerCommand: "sudo /etc/init.d/tomcat-tomcat1 start",
    stopServerCommand: "sudo /etc/init.d/tomcat-tomcat1 stop"
)
server1Deployer.deploy(artifacts.findAll{!it.name.contains("geoserver")})

def geoserverArtifact = artifacts.find{it.name.contains("geoserver")}
```

## Une réponse efficace à la directive INSPIRE

```
if (geoserverArtifact != null) {  
    def geoserverSSH = ssh.changeHost("server2")  
    def geoserverDeployer = tomcat1Deployer.copy(ssh: geoserverSSH)  
    geoserverDeployer.deploy()  
}
```

The above code is `trunk/server-deploy/exampleDeployScript.groovy` and has many comments describing the purpose of each line.

To reiterate. The server-deploy module provides a way to very simply write deploy scripts for deploying the entire system to one or more servers. The design goals of this module are as follows:

- Provide a way to very simply write deploy scripts
- Be platform independent so the one script will work on any platform
- Require no additional installations other than maven and java
- Be extremely flexible so that it is easy to write script that deploy all wars to a single server even deploy the same artifact to several servers for scaling and the other artifacts to other servers.

Currently the server-deploy-support provides only rudimentary support but extra classes can be added to assist in writing deploy scripts for other types of server configurations.

### ***Random technical information***

#### ***Java SSL, Keystores and Truststores***

A Keystore stores a servers certificates and credentials and is used when a server wants to authenticate with another server. If you want a tomcat (for example) to have a certificate you need to create a Keystore and put the certificate into that Keystore. Often the certificates are in DEM format so you can use a script like: <https://github.com/jesseeichar/jvm-security-scripts/blob/master/ImportDem.java> or <https://github.com/jesseeichar/jvm-security-scripts/blob/master/ImportDem.scala> to convert the DEM and install it into a Keystore. Naturally you need a Keystore before you can install anything into one so you can create one using the: [https://github.com/jesseeichar/jvm-security-scripts/blob/master/create\\_empty\\_Keystore](https://github.com/jesseeichar/jvm-security-scripts/blob/master/create_empty_Keystore) script that (obviously) creates an empty Keystore.

That is all good, but for 2 servers to connect one server needs a certificate and the other server needs to trust that certificate. That is where Truststores come in. By default the JVM ships with a Truststore with the major certificate vendors so if your certificate was created by one of them then you are good. If not then you need to create a custom Truststore. You start out with an empty Keystore (see above script for creating that) then you can import a servers certificate into that Keystore using one of the scripts: <https://github.com/jesseeichar/jvm-security-scripts/blob/master/InstallCert.java> or <https://github.com/jesseeichar/jvm-security-scripts/blob/master/InstallCert.scala>. The scripts essentially query the target server for its certificate then install that certificate into the Truststore.

One major gotcha is that the certificate and hostname are tied together so if the server has multiple aliases you need to choose the one you will use.

### ***Utilisation***

Une réponse efficace à la directive INSPIRE

Une documentation est en cours de transposition sur le site, contactez-nous si vous désirez y contribuer !

## **Administrateur**

### **Documentation administrateur**

Cette section rassemble la documentation autour du projet geOrchestra pour les administrateurs de l'IDS.

### **Manuel utilisateur de GeoServer**

GeoServer est un serveur logiciel open source écrit en Java qui permet aux utilisateurs de partager et d'éditer des données géospatiales. Pensé pour permettre l'intéropérabilité, il publie des données à partir des formats majeurs de données spatiales en utilisant des standards ouverts.

Ce manuel utilisateur est un guide complet sur tous les aspects d'utilisation de GeoServer. Que vous soyez un novice ou un vétéran de ce logiciel, nous espérons que cette documentation vous sera d'une grande aide.

## **Débuter**

Cette section du guide utilisateur contient un aperçu rapide de GeoServer pour permettre à des nouveaux utilisateurs de réaliser des tâches communes rapidement et simplement.

### **Démarrage rapide de l'interface d'administration web**

L'interface web d'administration est une application web utilisée pour configurer tous les aspects de GeoServer, de l'ajout de donnée à la personnalisation de paramètres des services. L'interface web est accessible via le navigateur web <http://<host>:<port>/geoserver>. <http://localhost:8080/geoserver/web> dans une installation par défaut fonctionnant sur un hôte local.



Page de bienvenue

## **Authentification**

Afin de changer n'importe quels paramètres ou configurer des données, un utilisateur doit d'abord s'authentifier. Naviguer jusqu'au coin en haut à gauche pour vous identifier dans GeoServer. L'utilisateur et mot de passe par défaut est admin et geoserver. Ceux-ci peuvent être seulement changé en éditant le fichier security/users.properties dans le *data\_directory*.

A screenshot of the GeoServer login form. It features input fields for 'username' (containing 'admin') and 'password' (containing a series of dots). There is a 'Remember me' checkbox and a green 'Login' button with a user icon.

Authentification

## **Serveur**

La section :guilabel: *Serveur* de l'admin web propose un accès aux informations d'environnement de GeoServer. C'est une combinaison d'outils de configuration et de diagnostic utile pour débuguer. La page *Status du serveur* offre un résumé des

Une réponse efficace à la directive INSPIRE

paramètres de configuration du serveur et du status en cours.

Server Status		
Summary of server configuration and status		
	Action	
Locks	0	<a href="#">Free locks</a>
Connections	6	
Memory Usage	28 MB	<a href="#">Free memory</a>
JVM Version	Sun Microsystems Inc.: 1.7.0-internal (Java HotSpot(TM) Server VM)	
Native JAI	false	
Native JAI ImageIO	false	
JAI Maximum Memory	377 MB	
JAI Memory Usage	0 KB	<a href="#">Free memory</a>
JAI Memory Threshold	75.0	
Number of JAI Tile Threads	8	
JAI Tile Thread Priority	5	
Update Sequence	35	
Resource Cache		<a href="#">Clear</a>
Configuration and catalog		<a href="#">Reload</a>
GeoServer		
Timestamps		
GeoServer	Jul 14, 3:07 PM	
Configuration	Jul 14, 3:07 PM	
XML	Mar 14, 2:15 PM	

### Page status

La section *Information de contact* définie des informations de contact public dans le document Capabilities du serveur WMS.

## Contact Information

Set the contact information for this server.

**Contact**

Claudius Ptolomaeus

**Organization**

The ancient geographers INC

**Position**

Chief geographer

**Address Type**

Work

**Address**

**City**

Alexandria

**State**

**ZIP code**

**Country**

*Page de contact*

La page *Paramétrages globaux* configure les paramètres de messages, log et proxy pour le serveur.

## Global Settings

Settings that apply to the entire server.

Verbose Messages

Verbose Exception Reporting

### Number of Decimals

8

### Character Set

UTF-8

### Proxy Base URL

### Logging Profile

DEFAULT\_LOGGING.properties  
VERBOSE\_LOGGING.properties  
PRODUCTION\_LOGGING.properties  
GEOTOOLS\_DEVELOPER\_LOGGING.properties  
GEOSEWER\_DEVELOPER\_LOGGING.properties

Log to StdOut

### Log Location

logs/geoserver.log

**Submit**

**Cancel**

*Page de paramétrage global*

La page *Configuration JAI* est utilisée pour configurer plusieurs paramètres JAI, utilisé à la fois dans les opérations WMS et WCS.

Une réponse efficace à la directive INSPIRE

## JAI Settings

Administer settings related to Java Advanced Imaging.

**Memory Capacity (0-1)**

0.5

**Memory Threshold (0-1)**

0.75

**Tile Threads**

7

**Tile Threads Priority**

5

Tile Recycling

Image I/O Caching

JPEG Native Acceleration

PNG Native Acceleration

Mosaic Native Acceleration

**Submit**

**Cancel**

### Paramètra JAI

La section *À propos de GeoServer* fournit des liens vers la documentation de GeoServer, la page principale et liste des bugs.

## Une réponse efficace à la directive INSPIRE

### About GeoServer

General information about GeoServer

#### GeoServer 2.0-RC1

The GeoServer project is a full transactional Java (J2EE) implementation of the OpenGIS Consortium's Web Feature Server specification and Web Coverage Server specification, with an integrated Web Map Server.

The documentation for this release is available online at the following link. The GeoServer wiki is used for the latest updates; please share your experiences, hints and tips with GeoServer there. The task tracker is the place to report feature requests and bugs. Also please take a moment to add yourself to the User Map to show your support for GeoServer.

- Documentation
- Wiki
- Bug Tracker

#### Section À propos

### Services

La section *Services* est pour les utilisateurs avancés qui ont besoin de configurer les protocoles demandés par GeoServer. La page Web Coverage Service (*WCS*) gère les informations de métadonnées, commune aux requêtes *WCS*, *WFS* et *WMS*. La page Web Feature Service (*WFS*) permet de configurer les features, niveaux de services et sortie GML. La page Web Map Service (*WMS*) définit les options raster et SVG.

### Données

Le lien *Données* amène directement à la page données avec les fonctionnalités d'édition, d'ajout et de suppression. Toutes les sous sections des types de données suivent un workflow similaire. De la même manière que ce que nous avons vu dans les exemples de *Styles* ci-dessous, la première page de chaque type de données affiche une page de visualisation des données sous forme de tableau.

### Styles

Manage the Styles published by GeoServer

- Add a new style
- Removed selected style(s)

The screenshot shows a list of styles in a web-based application. At the top, there are navigation buttons for navigating through the list of 22 items (Results 1 to 22 (out of 22 items)). A search bar is also present. The list itself has a header row with a checkbox and the title 'Style Name'. Below this, there are 22 items, each with a checkbox and a style name: 'burg', 'giant\_polygon', 'capitals', 'simple\_streams', 'pophatch', 'restricted', 'tiger\_roads', 'poly\_landmarks', 'green', and 'rain'. The style names 'burg', 'giant\_polygon', 'capitals', 'simple\_streams', 'pophatch', 'restricted', 'tiger\_roads', 'poly\_landmarks', and 'green' are in blue, while 'rain' is in black.

Page de vue des styles

## Une réponse efficace à la directive INSPIRE

Chaque nom de type de données amène vers une page de configuration correspondante. Par exemple, tous les items listés sous Workspace, Store et Layer Name dans la page de visualisation des *Couches*, amène vers sa page de configuration correspondante.

### Layers

Manage the layers being published by GeoServer

- Add a new resource
- Remove selected resources

Results 1 to 10 (out of 19 items)						Search
Type	Workspace	Store	Layer Name	Enabled?	Native SRS	
	nurc	arcGridSample	Arc_Sample		EPSG:4326	
	nurc	img_sample2	Pk50095		EPSG:32633	
	nurc	mosaic	mosaic		EPSG:4326	
	nurc	worldImageSample	Img_Sample		EPSG:4326	
	sf	sf	archsites		EPSG:26713	
	sf	sf	bugsites		EPSG:26713	
	sf	sf	restricted		EPSG:26713	
	sf	sf	roads		EPSG:26713	
	sf	sf	streams		EPSG:26713	
	sf	sfdem	sfdem		EPSG:26713	

### Vues des couches

Dans le panneau de visualisation des types, il y a trois façons différentes de localiser un type de données--par classement, par recherche et par l'ascenseur.

Pour classer alphabétiquement un type de donnée, cliquez sur l'en-tête de la colonne.

Style Name	Style Name
<input type="checkbox"/> burg	<input type="checkbox"/> burg
<input type="checkbox"/> giant_polygon	<input type="checkbox"/> capitals
<input type="checkbox"/> capitals	<input type="checkbox"/> che_bless
<input type="checkbox"/> simple_streams	<input type="checkbox"/> concat
<input type="checkbox"/> poatch	<input type="checkbox"/> dem
<input type="checkbox"/> restricted	<input type="checkbox"/> flags
<input type="checkbox"/> tiger_noids	<input type="checkbox"/> giant_polygon
<input type="checkbox"/> poly_landmarks	<input type="checkbox"/> grass
<input type="checkbox"/> green	<input type="checkbox"/> green
<input type="checkbox"/> rain	<input type="checkbox"/> line

Sur la gauche une colonne non classée ; sur la droite une colonne classée.

Pour la recherche simple, entrez le critère de recherche dans la boîte de recherche et taper entrer.

Results 1 to 1 (out of 1 matches from 7 items)						top
Workspace Name						
<input type="checkbox"/> topp						
<<	<	1	2	>	>>	Results 1 to 1 (out of 1 matches from 7 items)

Résultats de la recherche pour la requête "top".

Pour naviguer à travers les pages des types de données, utilisez le bouton flèche en bas et en haut du tableau.

						Results 1 to 25 (out of 27 items)
--	--	--	--	--	--	-----------------------------------

## Une réponse efficace à la directive INSPIRE

*Navigation dans la page des types de données.*

Comme nous l'avons vue dans l'exemple *Stores* ci-dessous, les boutons pour ajouter et supprimer les types de données peuvent être trouvés en haut de la page de visualisation.

## Stores

Manage the stores providing data to GeoServer

 [Add new Store](#)

 [Remove selected Stores](#)

*Boutons pour ajouter et supprimer des Stores.*

Pour ajouter de nouvelle donnée, sélectionnez le bouton *Ajouter*, et suivez les questions spécifique au type de donnée. Pour supprimer un type de données, cliquez sur la case à cocher du type de données correspondant et sélectionnez le bouton *Supprimer* (plusieurs types de données, du même type, peut être coché pour une suppression multiple).

## Stores

Manage the stores providing data to GeoServer

 [Add new Store](#)

 [Remove selected Stores](#)

Results 1 to 9 (out of 9 items)				
Type	Workspace	Store Name	Enabled?	
 	nurc	arcGridSample		
 	nurc	img_sample2		
 	nurc	mosaic		
<input checked="" type="checkbox"/>  	nurc	worldImageSample		
 	sf	sfdem		
 	sf	sf		
<input checked="" type="checkbox"/>  	tiger	nyc		
 	topp	states_shapefile		
 	topp	taz_shapes		

*Stores coché pour suppression.*

## Démos

La page *Démos* contient des liens vers des requêtes d'exemple de services WMS, WCS et WFS pour GeoServer ainsi qu'un lien listant toutes les informations de SRS connus de GeoServer. Vous n'avez pas besoin d'être identifier dans GeoServer pour visualiser cette page.

## GeoServer Demos

Collection of GeoServer demo applications and tools

- [Demo requests](#) Example requests for GeoServer (using the TestServlet).
- [SRS List](#) List of all SRS known to GeoServer

*Page de démos*

### Prévisualisation des couches

La page *Prévisualisation des couches* fournit des visualisations de couches pour plusieurs formats de sorties, incluant les formats OpenLayers et KML. Cette page aide à vérifier visuellement et à explorer la configuration d'une couche particulière.

#### Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

Type	Name	Title	Common Formats	All Formats
	nurc:Arc_Sample	A sample ArcGrid file	OpenLayers KML	Select one
	nurc:Pk50095	Pk50095 is a A raster file accompanied by a spatial data file	OpenLayers KML	Select one
	nurc:mosaic	Sample PNG mosaic	OpenLayers KML	Select one
	nurc:Img_Sample	North America sample imagery	OpenLayers KML	Select one
	sf:archsites	Spearfish archeological sites	OpenLayers KML GML	Select one
	sf:bugsites	Spearfish bug locations	OpenLayers KML GML	Select one
	sf:restricted	Spearfish restricted areas	OpenLayers KML GML	Select one
	sf:roads	Spearfish roads	OpenLayers KML GML	Select one
	sf:streams	Spearfish streams	OpenLayers KML GML	Select one
	sf:sfdem	sfdem is a Tagged Image File Format with Geographic information	OpenLayers KML	Select one
	tiger:poi	Manhattan (NY) points of interest	OpenLayers KML GML	Select one

*Page de prévisualisation des couches.*

Chaque ligne de couche comprend un *type*, *nom*, *titre*, et les formats disponibles pour la visualisation. *Nom* référence au nom du Workspace et de la couche, tandis que *Titre* se réfère à une brève description configurée dans le panneau [Éditez les données d'une couche](#). *Formats communs* inclus les rendus OpenLayers et KML, tandis que *Tous les formats* inclus des formats de rendus additionnels pour des utilisations plus particulière ou partager les données.

Type	Name	Title	Common Formats	All Formats
	nurc:Arc_Sample	A sample ArcGrid file	OpenLayers KML	Select one

*Ligne de visualisation d'une couche unique.*

### Ajouter un Shapefile

Ce tutorial présente les étapes de publication d'un Shapefile avec GeoServer.

#### Note

Ce tutorial suppose que GeoServer fonctionne sur <http://localhost:8090/geoserver/web>.

#### Commencer

#. Téléchargez le fichier *nyc\_roads.zip*. Ce fichier contient un shapefile des routes de la ville de New York qui sera utilisé dans ce tutorial.

1. Décompressez le fichier *nyc\_roads.zip*. Le répertoire extrait est constitué des quatres fichiers suivantes :

```
nyc_roads.shp  
nyc_roads.shx  
nyc_roads.dbf  
nyc_roads.prj
```

2. Déplacer le répertoire *nyc\_roads* dans <GEO SERVER DATA DIR>/data où GEO SERVER DATA DIR est la racine du répertoire données de GeoServer. Si aucun changement n'a été fait dans la structure du fichier de GeoServer, le chemin devrait être *geoserver/data\_dir/data/nyc\_roads*.

#### Créer un nouvel Workspace

La première étape est de créer un workspace pour le Shapefile. Le workspace est un conteneur utilisé pour grouper des couches semblables.

1. Dans un navigateur allez sur <http://localhost:8080/geoserver/web>.
2. Identifiez vous dans GeoServer comme décrit dans le démarrage rapide *Authentification*.
3. Naviguez dans *Données»Workspaces*.

#### Workspaces

The screenshot shows the 'Manage Geoserver workspaces' page. At the top, there are two buttons: 'Add new workspace' (green) and 'Remove selected workspace(s)' (red). Below these are navigation buttons ('<<', '<', '>', '>>') and a search bar. A message 'Results 1 to 7 (out of 7 items)' is displayed. The main area lists workspaces with checkboxes next to them. The listed workspaces are: sf, topp, it.geosolutions, sde, nurc, tiger, and cite. Each workspace name is preceded by a small icon representing its type (e.g., a map for sf).

Une réponse efficace à la directive INSPIRE

*page Workspaces*

4. Pour créer un nouvel workspace sélectionnez le bouton *Ajoutez un nouvel workspace*. On vous demandera d'entrer un *Nom* et l'*URI du Namespace* du workspace.

## New Workspace

Configure a new workspace

**Name**

**Namespace URI**

The namespace uri associated with this workspace

**Submit**

**Cancel**

*Configurer un nouvel Workspace*

5. Entrez le nom `nyc_roads` et `http://opengeo.org/nyc_roads` pour l'URI. Un nom de workspace est un nom décrivant votre projet et ne peut excéder dix caractères ou contenir des espaces. L'URI (Uniform Resource Identifier) du Namespace, est typiquement l'URL associée à votre projet, avec peut-être un identifiant différent.

## New Workspace

Configure a new workspace

**Name**

  
`nyc_roads`

**Namespace URI**

  
`http://opengeo.org/nyc_roads`

The namespace uri associated with this workspace

**Submit**

**Cancel**

*Workspace des routes de la ville de New York*

6. Cliquez sur le bouton *Soumettre*. GeoServer rajoutera le workspace `nyc_roads` en bas de la liste des Workspace.

Une réponse efficace à la directive INSPIRE

## Créer un Store

1. Naviguer vers *Data Stores*.
2. Dans le but d'ajouter les données nyc\_roads, nous devons créer un nouveau store. Cliquez sur le bouton *Ajouter une nouveau Store*. Vous serez dirigé vers une liste de types de données que GeoServer peut gérer.

### New data source

Choose the type of data source you wish to configure

#### Vector Data Sources

- Directory of spatial files** - Takes a directory of spatial data files and exposes it as a data store
- PostGIS NG** - PostGIS Database
- PostGIS NG (JNDI)** - PostGIS Database (JNDI)
- Properties** - Allows access to Java Property files containing Feature information
- Shapefile** - ESRI(tm) Shapefiles (\*.shp)
- Web Feature Server** - The WFSDataStore represents a connection to a Web Feature Server. This connection provides access to the Features published by the server, and the ability to perform transactions on the server (when supported / allowed).

#### Raster Data Sources

- ArcGrid** - Arc Grid Coverage Format
- GeoTIFF** - Tagged Image File Format with Geographic information
- Gtopo30** - Gtopo30 Coverage Format
- ImageMosaic** - Image mosaicking plugin
- WorldImage** - A raster file accompanied by a spatial data file

### Sources de données

3. Puisque que nyc\_roads est un shapefile, sélectionnez *Shapefile : ESRI(tm) Shapefiles (.shp)*.
4. Sur la page *Nouveau source de données vecteur* commencez par configurer les *Informations de base du Store*. Sélectionnez le workspace nyc\_roads à partir de la liste déroulante, entrez NYC Roads pour le nom puis entrez une brève description, comme Routes de la ville de New York.
5. Dans la partie *Paramètres de connexions* définissez la localisation du shapefile--file:data/nyc\_roads/nyc\_roads.shp.

New Vector Data Source

Shapefile  
ESRI Shapefiles (\*.shp)

**Basic Store Info**

Workspace  
nyc\_roads

Data Source Name  
NYC Roads

Description  
Roads in New York City

Enabled

**Connection Parameters**

url  
file:data/nyc\_roads/nyc\_roads.shp

namespace  
http://opengeo.org/nyc\_roads

create spatial index

charset  
ISO-8859-1

memory mapped buffer

### Information des données et paramètres de nyc\_roads

6. Pressez Sauver. Vous serez redirigé vers la page *Choix de la nouvelle couche* pour configurer la couche nyc\_roads.

## Configuration de couche

## Une réponse efficace à la directive INSPIRE

1. Sur la page *Choix de la nouvelle couche*, sélectionnez le nom de la couche nyc\_roads.

### New Layer chooser

Here is a list of resources contained in the store 'NYC Roads'. Click on the layer you wish to configure

<<	<	I	>	>>	Results 1 to 1 (out of 1 items)	<input type="text"/> Search
Layer with namespace and prefix						Published
nyc_roads						
<<	<	I	>	>>	Results 1 to 1 (out of 1 items)	

### Choix de la nouvelle couche

2. La configuration suivante définit les données les paramètres de publication pour une couche. Entrez un *Titre* court et un *Abstract* pour le shapefile nyc\_roads.

### nyc\_roads:nyc\_roads

Configure the resource and publishing information for the current layer

Data	Publishing
<b>Basic Resource Info</b>	
Name	<input type="text" value="nyc_roads"/>
Title	<input type="text" value="Subset of NYC roads"/>
Abstract	<input type="text" value="Shapefile of NYC roads"/>

### Information de base des ressources pour un Shapefile

3. Générez les *limites* du shapefile en cliquant sur *Calcul à partir des données* puis *Calcul à partir des limites natives*.

Une réponse efficace à la directive INSPIRE

### Bounding Boxes

#### Native Bounding Box

Min X	Min Y	Max X	Max Y
984,018.166	207,673.095	991,906.497	219,622.54

[Compute from data](#)

#### Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
-74.001	40.737	-73.972	40.769

[Compute from native bounds](#)

*Générer la Bounding Box*

4. Définissez le *style* du Shapefile en allant sur l'onglet *Publication*.
5. Puis sélectionnez *ligne* à partir de la liste *Style par défaut*.

## nyc\_roads:nyc\_roads

Configure the resource and publishing information for the current layer

[Data](#) [Publishing](#)

### Basic Settings

#### Name

nyc\_roads

- StyleInfoImpl[cite\_lakes]
- StyleInfoImpl[concat]
- StyleInfoImpl[dem]
- StyleInfoImpl[flags]
- StyleInfoImpl[giant\_polygon]
- StyleInfoImpl[grass]
- StyleInfoImpl[green]
- StyleInfoImpl[line]**
- StyleInfoImpl[medford\_buildings]
- StyleInfoImpl[medford\_citylimits]
- StyleInfoImpl[medford\_parks]
- StyleInfoImpl[medford\_streets]
- StyleInfoImpl[medford\_zoning]
- StyleInfoImpl[poi]
- StyleInfoImpl[point]
- StyleInfoImpl[poly\_landmarks]
- StyleInfoImpl[polygon]
- StyleInfoImpl[pop hatch]
- StyleInfoImpl[population]
- StyleInfoImpl[rain]
- StyleInfoImpl[rule]

*Sélection du style par défaut.*

6. Terminer la configuration de vos données et leur publication en descendant en bas de la page et en cliquant sur *Sauver*.

### Prévisualiser la couche

## Une réponse efficace à la directive INSPIRE

1. Afin de vérifier que la couche nyc\_roads est publiée nous allons la prévisualiser. Naviguez sur *Prévisualiser la couche* et chercher le lien de nyc\_roads:nyc\_roads.

### Layer Preview

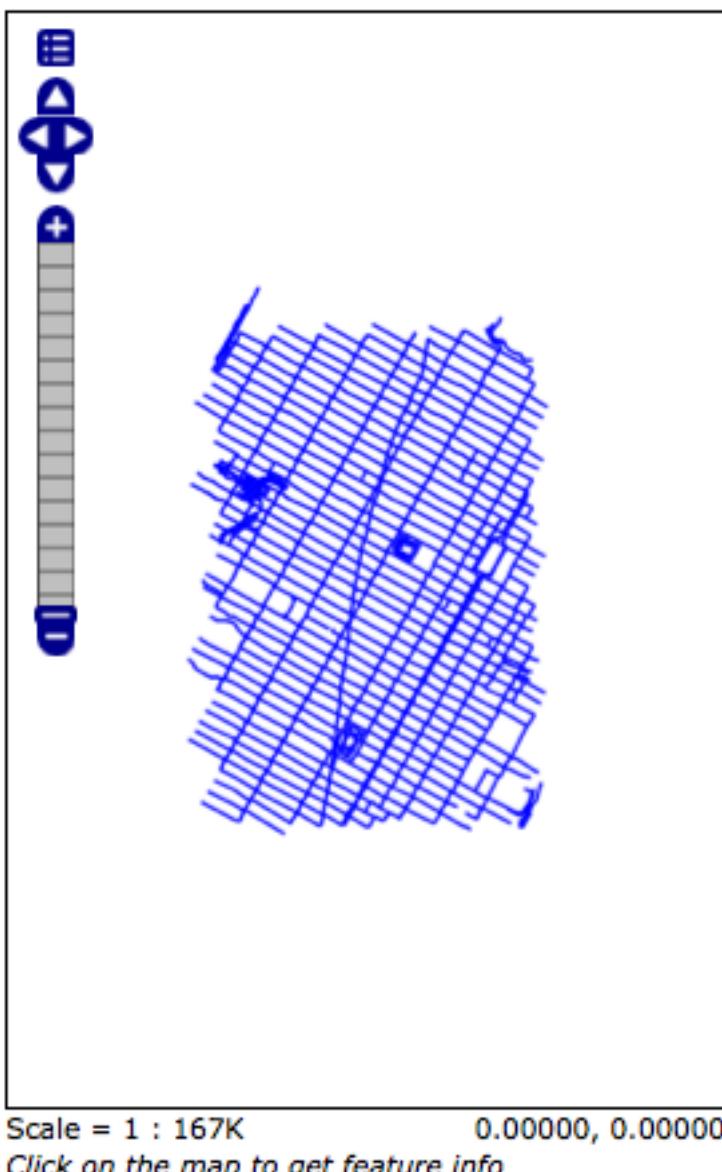
List of all layers configured in GeoServer and provides previews in various formats for each.

Results 1 to 25 (out of 26 items)			
Type	Name	Title	Common Formats
All Formats			
grid	nurc:Arc_Sample	A sample ArcGrid file	OpenLayers KML Select one
grid	nurc:Img_Sample	North America sample imagery	OpenLayers KML Select one
grid	nurc:Pk50095	Pk50095 is a A raster file accompanied by a spatial data file	OpenLayers KML Select one
grid	nurc:mosaic	Sample PNG mosaic	OpenLayers KML Select one
<input checked="" type="checkbox"/>	nyc_roads:nyc_roads	Subset of NYC roads	OpenLayers KML GML Select one

### Prévisualiser la couche

2. Cliquez sur le lien *OpenLayers* sur la colonne *Formats communs*.
3. Succès ! Une carte OpenLayers doit se charger avec le style par défaut de la ligne.

Une réponse efficace à la directive INSPIRE



Carte OpenLayers de nyc\_roads

#### Ajouter une table PostGIS

Ce tutorial montre les étapes pour publier une table PostGIS avec GeoServer.

#### Note

Ce tutorial suppose que GeoServer fonctionne sur <http://localhost:8090/geoserver/web>.

### Note

Ce tutorial suppose que PostGIS a été précédemment installé sur le système.

### Commencer

1. Téléchargez le fichier `nyc_buildings.zip`. Il contient un dump PostGIS d'un sous jeu de données des bâtiments de la ville de New York qui sera utilisé dans ce tutorial.
2. Créez une base de données PostGIS nommée "nyc". Cela peut être réalisé avec les commandes suivantes :

```
createdb -T template_postgis nyc
```

si l'installation de PostGIS n'a pas défini le modèle "postgis\_template" alors la séquence suivante fera la même chose :

```
...
```

3. Décompressez `nyc_buildings.zip` quelque part sur le système de fichier. Vous obtiendrez un fichier `nyc_buildings.sql`.
4. Importez `nyc_buildings.sql` dans la base nyc:

```
psql -f nyc_buildings.sql nyc
```

### Créer un nouveau store

La première étape est de créer un \* dataStore\* pour la base de données PostGIS "nyc". Le dataStore dira à GeoServer comment se connecter à la base de données.

1. Dans un navigateur web allez sur <http://localhost:8080/geoserver>.
2. Naviguez vers *Data▶Stores*.

## New data source

Choose the type of data source you wish to configure

### Vector Data Sources

- Directory of spatial files - Takes a directory of spatial data files and exposes it as a data store
- PostGIS NG - PostGIS Database
- PostGIS NG (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (\*.shp)
- Web Feature Server - The WFSDataStore represents a connection to a Web Feature Server. This connection provides access to the Features published by the server, and the ability to perform transactions on the server (when supported / allowed).

### Raster Data Sources

- ArcGrid - Arc Grid Coverage Format
- GeoTIFF - Tagged Image File Format with Geographic information
- Gtopo30 - Gtopo30 Coverage Format
- ImageMosaic - Image mosaicking plugin
- WorldImage - A raster file accompanied by a spatial data file

*Ajouter un nouveau dataStore.*

3. Créez un nouveau dataStore en cliquant sur le lien PostGIS NG.
4. En gardant le Workspace par défaut entrez le nom et la description dans *informations basiques du store*.

### Basic Store Info

#### Workspace

#### Data Source Name

#### Description

 Enabled

*Information basique du Store*

5. Définissez les paramètres de connexion de la base PostGIS.

dbtype	postgisng
host	localhost
port	5432
database	nyc
schema	public
user	postgres
passwd	entrez le mot de passe de postgres
validate connections	activez avec la case à cocher

### Note

Les paramètres spécifiques **username** et **password** de l'utilisateur qui a créé la base de données. En fonction de la manière dont PostgreSQL est configuré le paramètre peut ne pas être nécessaire.

#### Connection Parameters

**dbtype**

postgisng

**host**

localhost

**port**

5432

**database**

nyc\_buildings

**schema**

public

**user**

postgres

**passwd**

\*\*\*\*\*

**namespace**

<http://www.opengeospatial.net/cite>

**max connections**

10

**min connections**

1

**fetch size**

1000

**Connection timeout**

20

validate connections

Loose bbox

preparedStatements

Paramètres de connexion

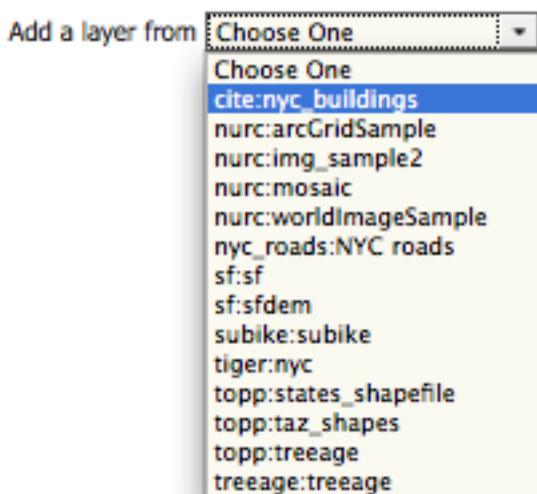
Une réponse efficace à la directive INSPIRE

6. Cliquez sur le bouton Sauver.

### **Configuration de la couche**

1. Naviguez vers *Données>Couches*.
2. Sélectionnez le bouton *Ajoutez une nouvelle ressource*.
3. À partir de la liste déroulante *Choix de la nouvelle couche*, sélectionnez *cite:nyc\_buildings*.

## New Layer chooser



*Sélection de la nouvelle couche dans la liste déroulante*

4. Dans la ligne de la couche résultante, sélectionnez le nom de la couche *nyc\_buildings*.

The screenshot shows a search interface with a search bar containing "<< < 1 > >> Results 0 to 0 (out of 0 items)". Below the search bar is a green header bar with the text "Layer with namespace and prefix". Underneath the header is a list box containing the item "nyc\_buildings".

*Nouvelle ligne de la couche*

5. La configuration suivante définit les paramètres de données et de publication pour une couche. Entrez les *Informations basiques de la ressource* pour *nyc\_buildings*.

## cite:nyc\_buildings

Configure the resource and publishing information for the current layer

Data Publishing

### Basic Resource Info

Name

nyc\_buildings

Title

NYC Buildings

Abstract

Subset of NYC buildings

Information basique de la ressource

6. Générez les *limites* de la table de la base de données en cliquant sur *Calcul à partir des données* puis sur *Calcul à partir des limites natives*.

### Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
983,837.625	207,499.469	991,858.938	218,794.359

Compute from data

Lat/Lon Bounding Box

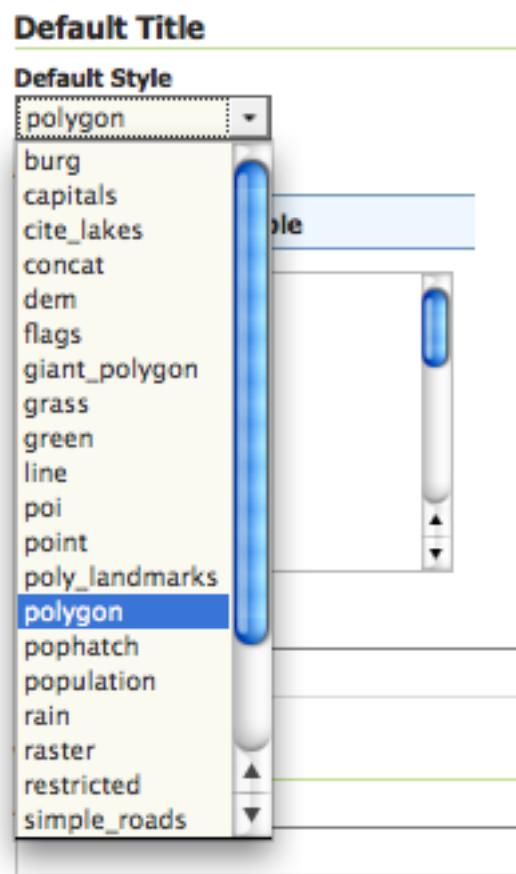
Min X	Min Y	Max X	Max Y
-74.002	40.736	-73.973	40.767

Compute from native bounds

Générez la Bounding Box

7. Définissez le *style* de la couche d'abord en allant sur l'onglet *Publication*.  
8. Puis sélectionnez *polygone* à partir de la liste déroulante *Style par défaut*.

Une réponse efficace à la directive INSPIRE



Sélection du style par défaut

9. Terminer la configuration de vos données et leur publication en descendant en bas de la page et en cliquant sur *Sauver*.

### Prévisualiser la couche

1. Afin de vérifier que la couche nyc\_building est publiée nous allons la prévisualiser. Naviguez sur *Prévisualiser la couche* et chercher le lien de cite:nyc\_building.

### Layer Preview

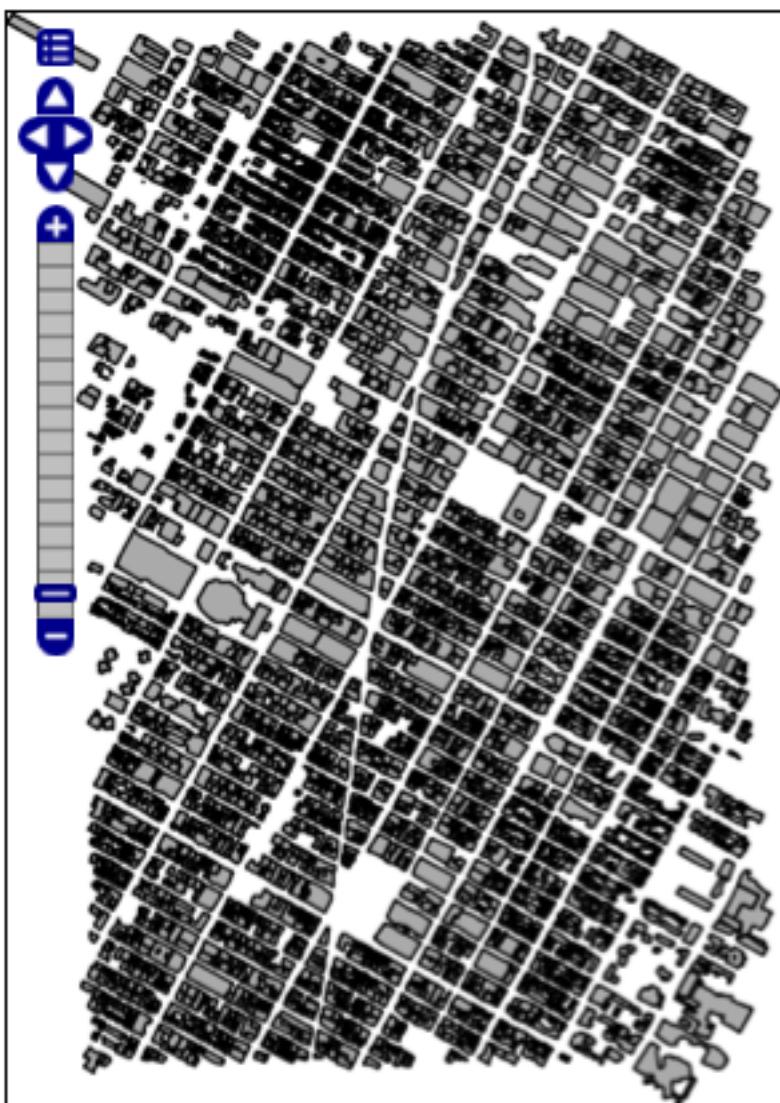
List of all layers configured in Geoserver and provides previews in various formats for each.

Results 1 to 25 (out of 27 items)			
Type	Name	Title	Common Formats
	cite:nyc_buildings	NYC Buildings	OpenLayers KML GML Select one

Prévisualiser la couche

2. Cliquez sur le lien *OpenLayers* sur la colonne *Formats communs*.
3. Succès ! Une carte OpenLayers doit se charger avec le style par défaut des polygones.

Une réponse efficace à la directive INSPIRE



*Carte OpenLayers de nyc\_buildings*

### **Styler une carte**

Quand un nouveau jeu de données est ajouté à GeoServer un style basic est assignée à la couche correspondante. Pour visualiser proprement la donnée un style spécifique à cette donnée doit être créé.

Ce tutorial montre les étapes pour créer un nouveau style dans GeoServer et fournir une introduction au style par le standard Styled Layer Descriptor (SLD).

## Note

Il est supposé que les tutoriaux [Ajouter un Shapefile](#) et [Ajouter une table PostGIS](#) ont été réalisés.

## Débuter

Avant de poursuivre avec ce tutorial il est *fortement recommandé* que la section [Introduction à SLD](#) soit lu en premier.

### Créer un nouveau style

#### Interface Web d'administration

L'interface web d'administration est un outil web pour configurer tous les aspects de GeoServer, de l'ajout de données au paramétrage des services. Elle est accessible via un navigateur web sur <http://localhost:8080/geoserver/web> dans une installation par défaut de GeoServer, bien que cette URL peut varier en fonction de l'installation de GeoServer.

#### Interface basique

Cette section introduira les concepts basiques de l'interface d'administration web (souvent abrégé en "admin web" dans le texte).

#### Page de bienvenue

Lorsque vous utilisez une installation normale, GeoServer démarrera un serveur web sur localhost accessible sur le port 8080 avec l'URL suivante :

`http://localhost:8080/geoserver/web`

## Note

Cette URL est dépendante de votre installation de GeoServer. Lors de l'utilisation de l'installation WAR, par exemple, l'URL dépendra de la configuration du conteneur.

Lorsque tout est correctement configuré, une page de bienvenue s'affiche dans votre navigateur.



#### Page de bienvenue

La page de bienvenue contient des liens vers différentes zones de configuration de GeoServer. La section *À propos de GeoServer* dans le menu Server fournit des liens externes vers la documentation de GeoServer, la home page et la liste des bugs. La page fournit également un accès pour s'identifier à la console de GeoServer. Cette mesure de sécurité

## Une réponse efficace à la directive INSPIRE

empeche les personnes non identifiées de modifier votre configuration de GeoServer. L'identification par défaut est admin et geoserver. Cela doit être changé en éditant le fichier security/users.properties dansz le répertoire *data\_directory*.



### Login

Sans être authentifié, le menu d'admin web amène vers la *Démo* et la *Prévisualisation des couches* de la console. La page *Démos* contient des liens utiles vers différentes pages d'informations, tandis que la page *Prévisualisation de couches* fournie des données spatiales dans différents formats de sortie.

Lorsque vous êtes identifié, des options supplémentaires sont présentes.

The screenshot shows the GeoServer administration interface. At the top, there's a login form with fields for 'admin' and 'password', a 'Remember me' checkbox, and a 'Login' button. Below the login is a 'Logout' button. The main content area has a header 'GeoServer'. On the left, a sidebar menu includes 'Server' (with Server Status, Contact Information, Global Settings, JAI Settings, About GeoServer), 'Services' (with WCS, WFS, WMS), 'Data' (with Workspaces, Stores, Layers, Layer Groups, Styles), 'Demos', and 'Layer Preview'. To the right, under 'Service Capabilities', it lists WCS (1.0.0, 1.1.1), WFS (1.0.0, 1.1.0), and WMS (1.1.1).

### Options supplémentaires lors de l'indentification

Les spécifications des configurations des Web Coverage Service (WCS), Web Feature Service (WFS), et Web Map Service (WMS) de Geoserver peuvent être accéder à partir de la page de bienvenu également. Voir la section sur services pour plus d'informations.

### Serveur

La section serveur de *Interface Web d'administration* propose un accès vers les informations d'environnement de GeoServer. C'est une combinaison d'outils de configuration et de diagnostique et peut être particulièrement utile pour le débogage.

### Status

The Server Status page offers a summary of server configuration parameters and run-time status. It is intended to assist diagnostics within a testing environment.

## Une réponse efficace à la directive INSPIRE



*Status Page*

### **Status Field Descriptions**

The following describes current status indicators

Option	Description
Locks	A WFS has the ability to lock features to prevent more than one person from updating the feature at one time. If data is locked, edits can be performed by a single WFS editor. When the edits are posted, the locks are released and features can be edited by other WFS editors. A zero in the locks field means all locks are released. If locks is non-zero, then pressing "free locks," releases all feature locks currently held by the server, and updates the field value to zero.
Connections	Refers to the numbers of vector stores, in the above case 4, that were able to connect.
Memory Usage	The amount of memory current used by GeoServer. In the above example, 55.32 MB of memory is being used. Clicking on the "Free Memory" button, cleans up memory marked for deletion by running the garbage collector.
JVM Version	Denotes which version of the JVM (Java Virtual Machine) is been used to power the server. Here the JVM is Apple Inc.: 1.5.0_16.
Native JAI	GeoServer uses <a href="#">Java Advanced Imaging (JAI)</a> framework for image rendering and coverage manipulation. When properly installed (true), JAI makes WCS and WMS performance faster and more efficient.
Native JAI ImageIO	GeoServer uses <a href="#">JAI Image IO (JAI)</a> framework for raster data loading and image encoding. When properly installed (true), JAI Image I/O makes WCS and WMS performance faster and more efficient.
JAI Maximum Memory	Expresses in bytes the amount of memory available for tile cache, in this case 33325056 bytes. The JAI Maximum Memory value must be between 0.0 and {0}

## Une réponse efficace à la directive INSPIRE

JAI Memory Usage	Run-time amount of memory is used for the tile cache. Clicking on the "Free Memory" button, clears available JAI memory by running the tile cache flushing.
JAI Memory Threshold	Refers to the percentage, e.g. 75, of cache memory to retain during tile removal. JAI Memory Threshold value must be between 0.0 and 100.
Number of JAI Tile Threads	The number of parallel threads used by the scheduler to handle tiles.
JAI Tile Thread Priority	Schedules the global tile scheduler priority. The priority value is defaults to 5, and must fall between 1 and 10.
Update Sequence	Refers to the number of times (60) the server configuration has been modified.
Resource cache	GeoServer does not cache data, but it does cache connection to stores, feature type definitions, external graphics, font definitions and CRS definitions as well. The "Clear" button forces those caches empty and makes GeoServer reopen the stores and re-read image and font information, as well as the custom CRS definitions stored in <code> \${GEOSEVER_DATA_DIR}/user_projections/epsg.properties</code> .
Configuration and catalog	GeoServer keeps in memory all of its configuration data. If for any reason that configuration information has become stale (e.g., an external utility has modified the configuration on disk) the "Reload" button will force GeoServer to reload all of its configuration from disk.

### ***Timestamps Field Descriptions***

Option	Description
GeoServer	Currently a placeholder. Refers to the day and time of current GeoServer install.
Configuration	Currently a placeholder. Refers to the day and time of last configuration change.
XML	Currently a placeholder.

### ***Information sur le contact***

Les informations de contact sont utilisées dans le document de capacité du service WMS et est publiquement accessible. Nous recommandons de remplir ce formulaire avec les informations appropriées afin que les utilisateurs puissent vous contacter.

## Contact Information

Set the contact information for this server.

### Contact

Claudius Ptolomaeus

### Organization

The ancient geographers INC

### Position

Chief geographer

### Address Type

Work

### Address

### City

Alexandria

### State

### ZIP code

### Country

Page de contact

### Champs d'information du contact

Champs	Description
Contact	Information de contact pour le webmaster.
Organisation	Le nom de l'organisation auquel le contact est affilié.
Position	La position du contact dans l'organisation.
Type d'adresse	Le type d'adresse définie, comme postal.
Adresse	La rue de l'adresse.
Ville	La ville de l'adresse.
État	L'état ou la province de l'adresse.

Une réponse efficace à la directive INSPIRE

Code postal	Le code postal de l'adresse.
Pays	Le pays de l'adresse.
Téléphone	Le numéro de téléphone du contact.
Fax	Le numéro de fax du contact.
Email	L'adresse mail du contact.

### **Global Settings**

The Global Setting page configures messaging, logging, character and proxy settings for the entire server.

## Global Settings

Settings that apply to the entire server.

Verbose Messages

Verbose Exception Reporting

#### Number of Decimals

8

#### Character Set

UTF-8

#### Proxy Base URL

#### Logging Profile

DEFAULT\_LOGGING.properties  
VERBOSE\_LOGGING.properties  
PRODUCTION\_LOGGING.properties  
GEOTOOLS\_DEVELOPER\_LOGGING.properties  
GEOSEWER\_DEVELOPER\_LOGGING.properties

Log to StdOut

#### Log Location

logs/geoserver.log

**Submit**

**Cancel**

## Global Settings Page

### Global Setting Fields

**Verbose Messages:** When enabled, Verbose Messages tells GeoServer to return XML with newlines and indents. Because such XML responses contain a larger amount of data, and in turn requires a larger amount bandwidth we recommended this option only for testing purposes.

**Verbose Exception Reporting:** Instead of the one line error message, enabled Verbose Exception Reporting returns service exceptions with full Java stack traces. Verbose exception reporting writes to the GeoServer log file and offers one of the most useful configuration options for debugging.

**Number of Decimals:** Refers to the number of decimal places returned in a GetFeature response. Also useful in optimizing bandwidth.

**Character Set:**\* Specifies the global character encoding that will be used in XML responses. We recommend the default UTF-8 for most users but support all character sets listed on the [IANA Charset Registry](#), and have an available Java implementation.

**Proxy Base URL:** GeoServer can have the capabilities documents properly report a proxy. The Proxy Base URL field is the base URL seen beyond a reverse proxy.

**Logging Profile:** Corresponds to a log4j configuration file in GeoServer's data directory. (Apache [log4j](#) is a Java-based logging utility.) By default, there are five logging profiles set in GeoServer's configurations file; customized profiles can be added by editing the log4j file.

There are six logging levels used by log. They range from the least serious TRACE, through DEBUG, INFO, WARN, ERROR and finally the most serious, FATAL. The GeoServer logging profiles combine logging levels with specific server operations. The five pre-built logging profiles available on the global settings page are:

1. *Default Logging* provides a good mix of detail without being VERBOSE. Default logging enables INFO on all GeoTools and GeoServer levels, except certain (chatty) GeoTools packages which require WARN.
2. *Verbose Logging* provides much more detail by enables DEBUG level logging on GeoTools, GeoServer, and VFNY.
3. *Production Logging* is the most minimal logging profile, with only WARN enabled on all GeoTools and GeoServer levels. With such production level logging only problems are written to the log files.
4. *GeoTools Developer Logging* is a verbose logging profile that includes DEBUG information only on GeoTools. This developer profile is recommended for active debugging of GeoTools.
5. *GeoServer Developer Logging* is a verbose logging profile that includes DEBUG information on GeoServer and VFNY. This developer profile is recommended for active debugging of GeoServer.

**Log to StdOut:** In general, StdOut (Standard output) refers to where a program writes its output data. In GeoServer, the Log to StdOut checkbox enables logging to the text terminal which initiated the program, most often the console. If you are running GeoServer in a large J2EE container, you might not want your container-wide logs filled with GeoServer information. Un-checking this option will suppress most GeoServer logging, with only fatal exceptions still outputted to the console log.

**Log Location** Sets the written output location for the logs. A log location may be a directory or a file, and can be specified as an absolute path (e.g., C:GeoServerGeoServer.log) or a

Une réponse efficace à la directive INSPIRE

relative one (e.g., GeoServer.log). Relative paths are relative to the GeoServer data directory.

### **Paramétrage de GeoWebCache**

The GeoWebCache Settings page in the Server menu in the *Interface Web d'administration* shows some configuration options for GeoWebCache, a tile server that comes embedded by default inside GeoServer. For more information about this embedded version, please see the section on *geowebcache*.

## GeoWebCache Settings

Configure the global settings for the embedded GeoWebCache

[Go to the GWC Home Page](#)

[Go to the GWC Demos Page](#)

### WMS Integration

Enable direct WMS integration

### Disk Quota

Enable Disk Quota limits

**Compute cache usage based on a disk block size of:**

4096 Bytes

**Check if the cache disk quota is exceeded every:**

10 Seconds

(Quota never forced since server start up)

**Set maximum tile cache size:**

100 MiB ▾

 Used 0.08 out of 100.0MiB available.

**When forcing disk quota limits, remove first tiles that are:**

Least Frequently Used

Least Recently Used

**Submit**

**Cancel**

### ***Enable direct WMS integration***

GeoWebCache acts as a proxy between GeoServer and map client. By default, GeoWebCache has a separate endpoint from the GeoServer WMS. (See the section on *gwc\_using* for more details.)

## Une réponse efficace à la directive INSPIRE

Enabling direct WMS integration allows WMS requests served through GeoServer to be cached as if they were received and processed by GeoWebCache. This yields the flexibility of a WMS with the speed of a tile server. See the section on *gwc\_using* for more details about this feature.

### Disk quota

This section manages the disk usage for tiles saved with GeoWebCache.

By default, disk usage with GeoWebCache is unbounded, regardless of integration with the GeoServer WMS, so every tile served from GeoWebCache will be stored in the cache directory (typically the gwc directory inside the data directory). When direct WMS integration is enabled but disk quotas not enabled, every tile that is served through both the GeoServer WMS and GeoWebCache will be stored in the cache directory, which could cause disk capacity issues. Setting a disk quota allows disk usage to be constrained.

Option	Default value	Description
<i>Enable Disk Quota limits</i>	Off	Turns on the disk quota. When disabled, the cache directory will grow unbounded. When enabled, the disk quota will be set according to the options below.
<i>Compute cache usage based on a disk block size of</i>	4096 bytes	This field should be set equal to the disk block size of the storage medium where the cache is located.
<i>Check if the cache disk quota is exceeded every</i>	10 seconds	Time interval at which the cache is polled. Smaller values (more frequent polling) will slightly increase disk activity, but larger values (less frequent polling) might cause the disk quota to be temporarily exceeded.
<i>Set maximum tile cache size</i>	100 MiB (Mebibytes)	The maximum size for the cache. When this value is exceeded and the cache is polled, tiles will be removed according to the policy choice listed below. Note that the unit options are <b>mebibytes</b> (approx. 1.05MB), <b>gibibytes</b> (approx. 1.07GB), and <b>tebibytes</b> (approx. 1.10TB).
<i>When forcing disk quota limits, remove first tiles that are</i>	Least Frequently Used	Sets the policy for tile removal when the disk quota is exceeded. Options are <b>Least Frequently Used</b> (removes tiles based on how often the tile was accessed) or <b>Least Recently Used</b> (removes tiles based on date of last access).

### Note

See the [GeoWebCache documentation](#) for more about disk quotas.

When finished making changes, click *Submit*.

## Une réponse efficace à la directive INSPIRE

This section also shows how much disk space is being used compared to the disk quota size, as well as the last time (if any) the quota was reached.

### Links

This page contains links to the embedded GWC homepage (containing runtime statistics and status updates) and *gwc\_demo* where you can view all layers known to GeoWebCache and reload configuration.

### JAI

[Java Advanced Imaging](#) (JAI) is an image manipulation library built by Sun Microsystems and distributed with an open source license. [JAI Image I/O Tools](#) provides reader, writer, and stream plug-ins for the standard Java Image I/O Framework. Several JAI parameters, used by both WMS and WCS operations, can be configured in the JAI Settings page.

## JAI Settings

Administer settings related to Java Advanced Imaging.

### Memory Capacity (0-1)

0.5

### Memory Threshold (0-1)

0.75

### Tile Threads

7

### Tile Threads Priority

5

Tile Recycling

Image I/O Caching

JPEG Native Acceleration

PNG Native Acceleration

Mosaic Native Acceleration

[Submit](#)

[Cancel](#)

*JAI Settings*

### **Memory & Tiling**

When supporting large images it is efficient to work on image subsets without loading everything to memory. A widely used approach is tiling which basically builds a tessellation of the original image so that image data can be read in parts rather than whole. Since very often processing one tile involves surrounding tiles, tiling needs to be accompanied by a tile-caching mechanism. The following JAI parameters allow you to manage the JAI cache mechanism for optimized performance.

**Memory Capacity:** For memory allocation for tiles, JAI provides an interface called TileCache. Memory Capacity sets the global JAI TileCache as a percentage of the available heap. A number between 0 and 1 exclusive. If the Memory Capacity is smaller than the current capacity, the tiles in the cache are flushed to achieve the desired settings. If you set a large amount of memory for the tile cache, interactive operations are faster but the tile cache fills up very quickly. If you set a low amount of memory for the tile cache, the performance degrades.

**Memory Threshold:** Sets the global JAI TileCache Memory threshold. Refers to the fractional amount of cache memory to retain during tile removal. JAI Memory Threshold value must be between 0.0 and 1.0. The Memory Threshold visible on the [Status](#) page.

**Tile Threads:** JAI utilizes a TileScheduler for tile calculation. Tile computation may make use of multithreading for improved performance. The Tile Threads parameter sets the TileScheduler, indicating the number of threads to be used when loading tiles.

**Tile Threads Priority:** Sets the global JAI Tile Scheduler thread priorities. Values range from 1 (Min) to 10 (Max), with default priority set to 5 (Normal).

**Tile Recycling:** Enable/Disable JAI Cache Tile Recycling. If checked, Tile Recycling allows JAI to re-use already loaded tiles, with vital capability for performances.

**Image I/O Caching:** Enables/disable Image I/O Caching. When checked, indicates that raw tiles read from disk should be cached.

**Native Acceleration:** In order to improve the computation speed of image processing applications, the JAI comes with both Java Code and native code for many platform. If the Java Virtual Machine (JVM) finds the native code, then that will be used. If the native code is not available, the Java code will be used. Thus, the JAI package is able to provide optimized implementations for different platforms that can take advantage of each platform's capabilities.

**JPEG Native Acceleration:** Enables/disable JAI JPEG Native Acceleration. When checked, enables JPEG native code, which may speed performance, but compromise security and crash protection.

**PNG Native Acceleration:** Enables/disables JAI PNG Native Acceleration. When checked, enables PNG native code, which may speed performance, but compromise security and crash protection.

**Mosaic Native Acceleration:** In order to reduce the overhead of handling them, very large data sets are often split into smaller chunks and then combined to create an image mosaic. An example of this can be found in aerial imagery which is usually comprised of thousands and thousands of small images at very high resolution (order of cm). Both native and JAI implementations of mosaic are provided. When checked, Mosaic Native Acceleration use the native implementation for creating mosaics.

### **Services**

GeoServer serves data using standard protocols established by the [Open Geospatial Consortium](#) (OGC). Web Coverage Service (WCS) allows for requests of coverage data (rasters); Web Feature Service (WFS) allows for requests of geographical feature data

## Une réponse efficace à la directive INSPIRE

(vectors); and Web Map Service (WMS) allows for requests of images generated from geographical data.

This section of the Web Administration Interface allows for the configuration of these services as used by GeoServer.

### WCS

The Web Coverage Service (WCS) offers few options for changing coverage functionality. While various elements can be configured for WFS and WMS requests, WCS allows only metadata information to be edited. This metadata information, entitled *Service Metadata*, are elements common to WCS, WFS and WMS requests.

#### Service Metadata

Enable WCS

Strict CITE compliance

#### Maintainer

<http://jira.codehaus.org/secure/BrowseProject.jspa>

#### Online resource

<http://geoserver.sourceforge.net/html/index.php>

#### Title

Web Coverage Service

#### Abstract

This server implements the [WCS](#) specification 1.0 and 1.1.1, it's reference implementation of [WCS 1.1.1](#). All layers published by this service are available on [WMS](#) also.

#### Fees

NONE

#### Access Constraints

NONE

#### Current Keywords

WCS  
WMS  
GEO SERVER

[Remove selected](#)

#### New Keyword

*WCS Configuration page*

### Service Metadata

## Une réponse efficace à la directive INSPIRE

WCS, WFS, and WMS use common metadata definitions. These nine elements are briefly described in the following table. Though these field types are the same regardless of service, their values are not shared. As such, parameter definitions below refer to the respective service. For example, "Enable" on the WFS Service page, enables WFS service requests and has no effect on WCS or WMS requests.

Field	Description
Enabled	Specifies whether the respective services--WCS, WFS or WMS--should be enabled or disabled. When disabled, the respective service requests will not be processed.
Strict CITE compliance	When checked, enforces strict OGC Compliance and Interoperability Testing Initiative (CITE) conformance. Recommended for use when running conformance tests.
Maintainer	Name of the maintaining body
Online Resource	Defines the top-level HTTP URL of the service. Typically the Online Resource is the URL of the service "home page." (Required)
Title	A human-readable title to briefly identify this service in menus to clients. (Required)
Abstract	Provides a descriptive narrative with more information about the service.
Fees	Indicates any fees imposed by the service provider for usage of the service. The keyword NONE is reserved to mean no fees and fits most cases.
Access Constraints	Describes any constraints imposed by the service provider on the service. The keyword NONE is reserved to indicate no access constraints are imposed and fits most cases.
Keywords	List of short words associated with the service to aid in cataloging. searching.

## **WFS**

The Web Feature Service (WFS) page allows for configuration of features, service levels, and GML output.

The screenshot shows a configuration form for a Web Feature Service (WFS). It includes fields for setting the maximum number of features (1000000), returning bounding boxes, selecting the service level (Basic, Transactional, Complete, selected), and choosing GML2 and GML3 output styles (XML and SRS style).

Features	Maximum number of features 1000000
<input type="checkbox"/> Return bounding box with every feature	
Service Level	
<input type="radio"/> Basic	
<input type="radio"/> Transactional	
<input checked="" type="radio"/> Complete	
GML2	SRS style XML
GML3	SRS style URN

*WFS configuration options*

## **Service Metadata**

See the section on [Service Metadata](#).

## **Features**

## Une réponse efficace à la directive INSPIRE

The [Open Geospatial Consortium](#) (OGC) Web Feature Service (WFS) is a standard protocol for serving geographic features across the Web. Feature information that is encoded and transported using WFS includes both feature geometry and feature attribute values. Basic Web Feature Service (WFS) supports feature query and retrieval. Feature limits and bounding can be configured on the WFS page.

**Maximum number of features:** A WFS request can potentially contain a very large dataset that is impractical to download to a client, and/or too large for a client's renderer. Maximum number of features sets the global feature limit that a WFS GetFeature operation should generate, regardless of the actual number of query hits. Maximum feature limits are also available for feature types. The default number is 1000000.

**Return bounding box:** Includes in the GetFeature GML output, an auto-calculated bounds element on each feature type. Not typically enabled, as including bounding box takes up extra bandwidth.

### Service Levels

GeoServer is compliant with the full "Transactional Web Feature Server" (WFS-T) level of service as defined by the OGC. Specifying the WFS service level limits the capabilities of Geoserver while still remaining compliant. The WFS Service Level is an integer bitmask that indicates what WFS operations are "turned on." It defines the available operations and content at a service instance

**Basic:** The Basic service levels provides facilities for searching and retrieving feature data with the GetCapabilities, DescribeFeatureType and GetFeature operations. It is compliant with the OGC basic Web Feature Service. This is considered a READ-ONLY web feature service.

**Transactional:** In addition to all basic WFS operations, transactional service level supports transaction requests. A transaction request facilities the creation, deletion, and updating of geographic features in conformance with the OGC Transactional Web Feature Service (WFS-T).

**Complete:** Includes the LockFeature support to the suite of transactional level operations. LockFeature operations help resolve links between related resources by processing lock requests on one or more instances of a feature type.

### GML

Geography Markup Language (GML) is the XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet.

The older GML standard, [GML 2](#) encodes geographic information, including both spatial and non-spatial properties. GML3 extends GML2 support to 3D shapes (surfaces and solids) as well as other advanced facilities. GML3 is modular superset of GML2 that simplifies and minimizes the implementation size by allowing users to select out necessary parts. Additions in GML3 include support for complex geometries, spatial and temporal reference systems, topology, units of measure, metadata, gridded data, and default styles for feature and coverage visualization. GML3 is almost entirely backwards compatible with GML2.

WFS 1.1.0 requests return GML3 as the default GML and style a Spatial Reference System (SRS) is in the URN format. Meanwhile WFS 1.0.0 requests return GML2 as default and specify SRS in the XML or normal format. These formats effect the longitude/latitude (x/y) order of the returned data and are further described below.

**Normal:** Returns the typical EPSG number: EPSG:XXXX. This formats the geographic coordinates in longitude/latitude (x/y) order.

Une réponse efficace à la directive INSPIRE

**XML:** Returns a URL that identifies each EPSG code: <http://www.opengis.net/gml/srs/epsg.xml#XXXX>. This formats the geographic coordinates in longitude/latitude (x/y) order.

**URN:** WFS 1.1.1 only. Returns the colon delimited SRS formatting: <urn:x-ogc:def:crs:EPSG:XXXX>. This formats data in the traditional axis order for geographic and cartographic systems: latitude/longitude (y/x).

## WMS

The Web Map Service (WMS) page allows for configuration of raster rendering and SVG options.

The screenshot shows a configuration interface for a Web Map Service (WMS). It includes three main sections: 'Raster Rendering Options' (with a dropdown for 'Default Interpolation' set to 'Nearest neighbor'), 'Watermark Settings' (with checkboxes for 'Enable watermark' and 'Watermark URL', and fields for 'Watermark Transparency' and 'Watermark Position'), and 'SVG Options' (with checkboxes for 'SVG Predictor', 'Strik', and 'Enable Antialiasing'). At the bottom are 'Submit' and 'Cancel' buttons.

*WMS configuration options*

## Service Metadata

See the section on [Service Metadata](#).

## Raster Rendering Options

The Web Map Service Interface Standard (WMS) provides a simple way to request and serve geo-registered map images. During pan and zoom operations, WMS requests generate map images through a variety of raster rendering processes. Such image manipulation is generally called resampling, interpolation, or down-sampling. GeoServer supports three resampling methods that determine how cell values of a raster are outputted. These sampling methods--Nearest Neighbor, Bilinear Interpolation and Bicubic--are available on the Default Interpolation drop-down menu.

**Nearest Neighbor:** Uses the center of nearest input cell to determine the value of the output cell. Original values are retained and no new averages are created. Because image values stay exactly the same, rendering is fast but possibly pixelated from sharp edge detail. Nearest neighbor interpolation is recommended for categorical data such as land use classification.

**Bilinear** Determines the value of the output cell based by sampling the value of the four nearest cells by linear weighting. The closer an input cell, the higher its influence on the output cell value. Since output values may differ from nearest input, bilinear interpolation is recommended for continuous data like elevation and raw slope values. Bilinear interpolation takes about five times as long as nearest neighbor interpolation.

**Bicubic** Looks at the sixteen nearest cells and fits a smooth curve through the points to find the output value. Bicubic interpolation may both change the input value as well as place the output value outside of the range of input values. Bicubic interpolation is recommended for smoothing continuous data, but at significant costs to speed.

## Watermark Settings

Watermarking is the process of embedding an image into a map. Watermarks are usually used for branding, copyright and security measures. Configuring watermarking is done in the WMS watermark settings section.

**Enable Watermark:** Turns on watermarking. When checked, all maps will render with the same watermark. It is not currently possible to specify watermarking on a per-layer or per-feature basis.

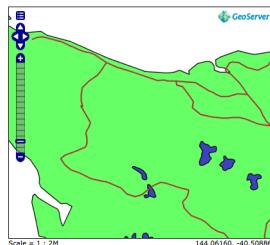
**Watermark URL:** This is the location of the graphic for the watermark. The graphic can be referenced as an absolute path (e.g., C:GeoServerwatermark.png), a relative one inside GeoServer's data directory (e.g., watermark.png), or a URL (e.g., <http://www.example.com/images/watermark.png>).

Each of these methods have their own advantages and disadvantages. When using an absolute or relative link, GeoServer keeps a cached copy of the graphic in memory, and won't continually link to the original file. This means that if the original file is subsequently deleted, GeoServer won't register it missing until the watermark settings are edited. Using a URL might seem more convenient, but it is more I/O intensive. GeoServer will load the watermark image for every WMS request. Also, should the URL cease to be valid, the layer will not properly display.

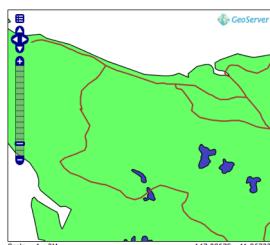
**Watermark Transparency:** Determines the opacity level of the watermark. Numbers range between 0 (opaque) and 100 (fully invisible).

**Watermark Position:** Specifies the position of the watermark relative to the WMS request. The nine options indicate which side and corner to place the graphic (top-left, top-center, top-right, etc). The default watermark position is bottom-right. Note that the watermark will always be displayed flush with the boundary. If extra space is desired, the graphic itself needs to change.

Because each WMS request renders the watermark, a single tiled map positions one watermark relative to the view window while a tiled map positions the watermark for each tile. The only layer specific aspect of watermarking occurs because a single tile map is one WMS request, whereas a tiled map contains many WMS requests. (The latter watermark display resembles Google Maps faint copyright notice in their Satellite imagery.) The following three examples demonstrate watermark position, transparency and tiling display, respectively.

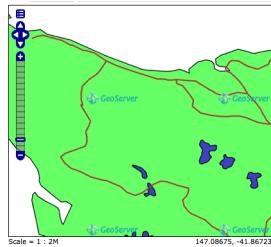


*Single tile watermark (aligned top-right, transparency=0)*



*Single tile watermark (aligned top-right, transparency=90)*

## Une réponse efficace à la directive INSPIRE



*Tiled watermark (aligned top-right, transparency=90)*

### **SVG Options**

The GeoServer WMS supports SVG (Scalable Vector Graphics) as an output format. GeoServer currently supports two SVG renderers available on the SVG producer drop down menu.

#### **SVG Producer:**

1. *Simple*: Simple SVG renderer. It has limited support for SLD styling, but is very fast.
2. *Batik*: Batik renderer (as it uses the Batik SVG Framework). It has full support for SLD styling, but is slower.

**Enable Anti-aliasing** Anti-aliasing is a technique for making edges appear smoother by filling in the edges of an object with pixels that are between the object's color and the background color. Anti-aliasing creates the illusion of smoother lines and smoother selections. Turning on anti-aliasing will generally make your maps look nicer, but will increase the size of the images returned, and will take a slight bit longer. Note that if you are overlaying the anti-aliased map on top of others it can sometimes backfire with transparencies, since it mixes with the colors behind and can create a "halo" effect.

### **Données**

Cette section est la section la plus grande et peut être la plus importante de l'interface d'administration web. Chaque sous-section amène directement à une page de type de données avec des possibilités d'ajout, d'édition et de suppression.

Comme vue dans l'exemple ci-dessous, la page de visualisation des données affiche un tableau des données indexées.

## Une réponse efficace à la directive INSPIRE

### Layers

Manage the layers being published by GeoServer

- [Add a new resource](#)
- [Remove selected resources](#)

Results 1 to 10 (out of 19 items)					
Type	Workspace	Store	Layer Name	Enabled?	Native SRS
	nurc	arcGridSample	Arc_Sample		EPSG:4326
	nurc	img_sample2	Pk50095		EPSG:32633
	nurc	mosaic	mosaic		EPSG:4326
	nurc	worldImageSample	Img_Sample		EPSG:4326
	sf	sf	archsites		EPSG:26713
	sf	sf	bugsites		EPSG:26713
	sf	sf	restricted		EPSG:26713
	sf	sf	roads		EPSG:26713
	sf	sf	streams		EPSG:26713
	sf	sfdem	sfdem		EPSG:26713

### Page de des couches

Pour ordonner alphabétiquement un type de données, cliquez sur l'en-tête de la colonne.

Style Name	Style Name
burg	burg
giant_polygon	capitals
capitals	cite_bless
simple_streams	concat
prostitch	dem
restricted	flags
tiger_roads	giant_polygon
poly_landmarks	grass
green	green
tan	line

Sur la gauche une colonne non ordonnées ; sur la droite une colonne ordonnée.

Pour la recherche simple du contenu des types de données, entrez le critère de recherche dans la boîte de recherche puis entrer. GeoServer recherchera les types de données qui sont pertinent et renverra une page de résultats.

Results 1 to 1 (out of 1 matches from 7 items)	
Workspace Name	top
topp	
Results 1 to 1 (out of 1 matches from 7 items)	

Résultats de recherche pour la requête "top".

Des détails spécifiques pour l'ajout, l'édition et la suppression des différents types de données sont présentés dans les sections suivantes.

### Espace de travail

Cette section permet de visualiser et configurer les espaces de travail. Analogue à un espace de nom, un espace de travail est un conteneur qui est utilisé pour organiser des objets. Dans GeoServer, un espace de travail est souvent utilisé pour regrouper des couches similaires ensemble. On se réfère souvent à des couches individuelles par leur nom d'espace de travail, deux points, puis leur entrepôt (ex. topp:states). Deux couches différentes peuvent

## Une réponse efficace à la directive INSPIRE

avoir le même nom du moment qu'ils sont dans deux espaces de travail différents (par exemple sf:states, topp:states).

### Workspaces

Manage GeoServer workspaces

 Add new workspace

 Remove selected workspace(s)

<< < I > >> Results 1 to 7 (out of 7 items)

 Search

**Workspace Name**

sf

topp

it.geosolutions

sde

nurc

tiger

cite

<< < I > >> Results 1 to 7 (out of 7 items)

*Page des espaces de travail*

### Éditer un espace de travail

Afin de voir les détails et d'éditer un espace de travail, cliquez sur son nom.

### Workspace topp

#### Namespace URI

The namespace uri associated with this workspace

**Save**

**Cancel**

*Espace nommé "topp"*

Un espace de travail consiste d'un nom et d'une URI d'espace de nom (Uniform Resource Identifier). Le nom de l'espace de travail a au maximum 10 caractères et ne peut pas contenir d'espace. Une URI est similaire à une URL, sauf qu'une URI ne nécessite pas de pointer vers un endroit sur le web, et doit seulement être un identifiant unique. Pour une URI d'espace de travail, nous recommandons d'utiliser une URL associée à votre projet, avec éventuellement un nom final d'identification, comme <http://www.openplans.org/topp> pour l'espace de travail "topp".

### Ajouter ou supprimer un espace de travail

Les boutons pour ajouter et supprimer un espace de travail peuvent être trouvé en haut de la page de visualisation des espaces.

Manage GeoServer workspaces  
 Add new workspace  
 Remove selected workspace(s)

Une réponse efficace à la directive INSPIRE

#### *Boutons pour ajouter et supprimer des espaces de travail*

Pour ajouter un espace de travail, sélectionnez le bouton *Ajouter un nouvel espace de travail*. On vous demandera d'entrer le nom de l'espace et son URI.

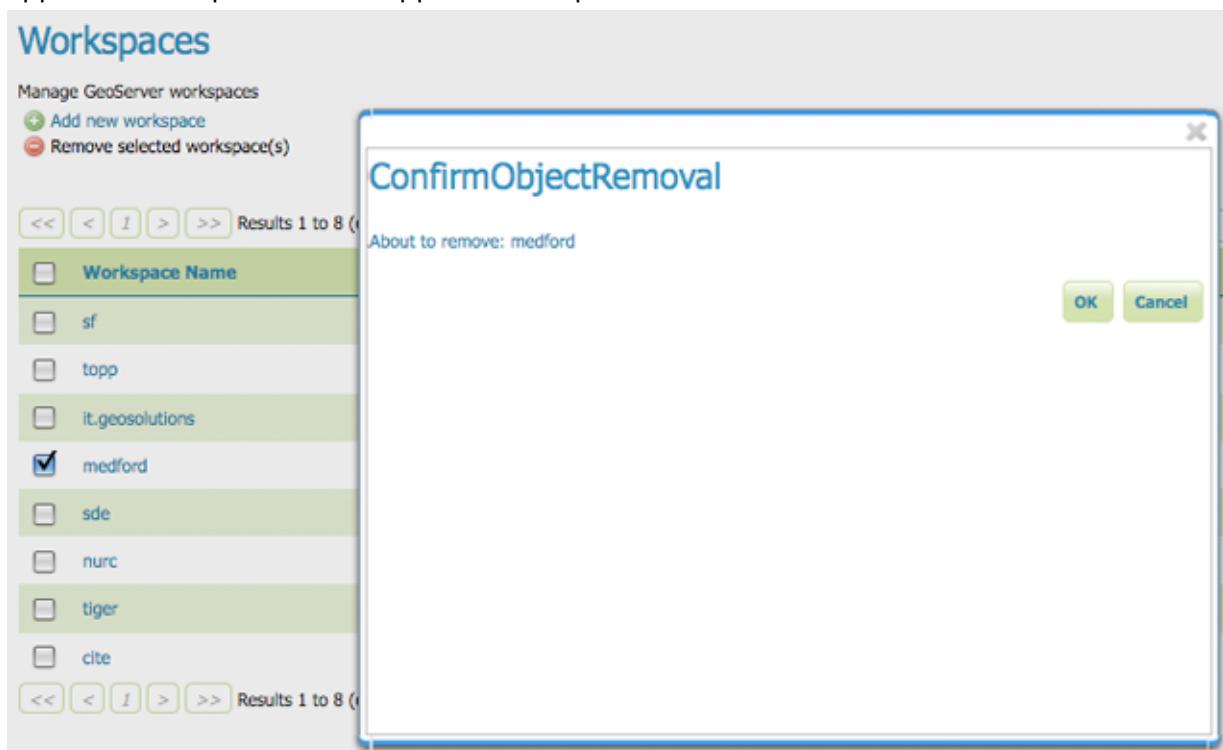
## New Workspace

Name

Namespace URI  
  
The namespace uri associated with this workspace

#### *Nouvel espace de travail avec un exemple*

Pour supprimer un espace de travail, cliquez sur la case à cocher correspondante de l'espace de travail. Comme pour le processus de suppression de couche, plusieurs espaces de travail peuvent être sélectionnés pour suppression dans une seule page de résultats. Cliquez sur le bouton *Supprimer les espaces de travail sélectionnés*. Vous devrez confirmer ou annuler la suppression. Cliquer sur *OK* supprimera l'espace de travail.



#### *Confirmation de la suppression de l'espace de travail*

## Entrepôt

Un entrepôt connecte une source de données qui contient des données raster ou vecteur. Une source de données peut être un fichier ou un groupe de fichier autant qu'une table dans une base de données, un fichier unique (comme un shapfile) ou un répertoire (telle que la

## Une réponse efficace à la directive INSPIRE

bibliothèque du Format de Produit Vecteur). L'entrepôt construit est utilisé afin que les paramètres de connexion soient définis qu'une seule fois, plutôt que pour chaque données à partir d'une source. Pour cela, il est nécessaire d'enregistrer un entrepôt avant le chargement des données.

### Stores

Manage the stores providing data to GeoServer

-  [Add new Store](#)
-  [Remove selected Stores](#)

Results 1 to 9 (out of 9 items)			
Type	Workspace	Store Name	Enabled?
 	nurc	arcGridSample	
 	nurc	img_sample2	
 	nurc	mosaic	
 	nurc	worldImageSample	
 	sf	sfdem	
 	sf	sf	
 	tiger	nyc	
 	topp	states_shapefile	
 	topp	taz_shapes	

### Vues des stores

Bien qu'il peut y avoir plusieurs formats potentiels pour une source de données, il y a seulement quatre types d'entrepôts. Pour les données raster, un entrepôt peut être un fichier. Pour les données vecteur, un entrepôt peut être un fichier, une base de données ou un serveur.

Icône du type	Description
	données raster dans un fichier
	données vecteur dans un fichier
	données vecteur dans une base de données
	serveur vectoriel (web feature server)

### Éditer un entrepôt

Afin de voir et d'édition un entrepôt, cliquez sur le nom de l'entrepôt. Le contenu exact de cette page dépendra du format choisi (voir la section sur *data* pour des informations sur les formats de données). Dans l'exemple plus bas nous avons le contenu de l'entrepôt nurc:ArcGridSample.

## Edit Raster Data Source

ArcGrid  
Arc Grid Coverage Format

### Basic Store Info

#### Workspace

#### Data Source Name

#### Description

 Enabled

### Connection Parameters

#### URL

### Éditer un entrepôt de données raster

Bien que les paramètres de connexion vont varier en fonction du format de données, certaines informations basiques sont communes à plusieurs formats. La liste déroulante des Espaces de noms liste tous les espace de noms enregistrés. On assigne l'entrepôt à l'espace de nom sélectionné (nurc). *Nom de la source de données* est le nom de l'entrepôt qui sera listé dans la page de visualisation. *Description* est optionnel et s'affiche seulement dans l'interface d'administration. *:guilabel:`Activé`* permet d'activer ou de désactiver l'entrepôt, avec toutes les données qui y est définie.

### Ajouter un entrepôt

Les boutons ajouter et supprimer un espace de nom peuvent être trouvé en haut de la page Entrepôts.

## Stores

Manage the stores providing data to GeoServer

 Add new Store

 Remove selected Stores

*Boutons pour ajouter et supprimer des entrepôts*

Pour ajouter un espace de nom, sélectionnez le bouton *Ajouter un nouvel entrepôt*. Une source de données vous sera demandée. GeoServer gère nativement plusieurs formats (et plus encore via les extensions). Cliquez sur la source de données appropriée pour continuer.

### New Store chooser

#### Vector Data Sources

-  Directory of spatial files - Takes a directory of spatial data files and exposes it as a data store
-  PostGIS NG - PostGIS Database
-  PostGIS NG (JNDI) - PostGIS Database (JNDI)
-  Properties - Allows access to Java Property files containing Feature information
-  Shapefile - ESRI(tm) Shapefiles (\*.shp)
-  Web Feature Server - The WFSDataStore represents a connection to a Web Feature Server. This connection provides access to perform transactions on the server (when supported / allowed).

#### Raster Data Sources

-  ArcGrid - Arc Grid Coverage Format
-  GeoTIFF - Tagged Image File Format with Geographic information
-  Gtopo30 - Gtopo30 Coverage Format
-  ImageMosaic - Image mosaicking plugin
-  WorldImage - A raster file accompanied by a spatial data file

*Choisir une source de données pour une nouvel entrepôt*

La page suivante configurera les entrepôts (l'exemple ci-dessous montre la page de configuration du raster ArcGrid). Cependant puisque les paramètres de connexion diffèrent d'une source de données à l'autre, le contenu exact de cette page dépendra du format spécifique de l'entrepôt. Lisez la section :ref:`data` sur les formats de données spécifiques.

Une réponse efficace à la directive INSPIRE

## Add Raster Data Source

ArcGrid  
Arc Grid Coverage Format

### Basic Store Info

#### Workspace

cite

#### Data Source Name

#### Description

Enabled

### Connection Parameters

#### URL

file:data/example.extension

**Save**

**Cancel**

*Page de configuration pour une source de données raster ArcGrid*

### **Supprimer un entrepôt**

Dans le but de supprimer un entrepôt, cliquez sur la case à cocher correspondant à l'entrepôt. Plusieurs entrepôts peuvent être sélectionnés pour des suppressions multiples.

## Une réponse efficace à la directive INSPIRE

### Stores

Manage the stores providing data to GeoServer



Type	Workspace	Store Name	Enabled?
	nurc	arcGridSample	
	nurc	img_sample2	
	nurc	mosaic	
<input checked="" type="checkbox"/>	nurc	worldImageSample	
	sf	sfdem	
	sf	sf	
<input checked="" type="checkbox"/>	tiger	nyc	
	topp	states_shapefile	
	topp	taz_shapes	

### Entrepôt sélectionné pour suppression

cliquez sur le bouton *Supprimer les entrepôts sélectionnés*. Vous devrez confirmer la suppression des données dans chaque entrepôt. Sélectionner *OK* supprimer les entrepôts et vous redirige vers la page des entrepôts principales.

Stores

Manage the stores providing data to GeoServer

<< < I > >> Results 1 to 9 (out of 9 items)

Type	Workspace	Store Name
	nurc	arcGridSample
	nurc	img_sample2
	nurc	mosaic
<input checked="" type="checkbox"/>	nurc	worldImageSample
	sf	sfdem
	sf	sf
<input checked="" type="checkbox"/>	tiger	nyc
	topp	states_shapefile
	topp	taz_shapes

About to remove: worldImageSample, nyc

The following objects will be removed:

- \* : Img\_Sample, poly\_landmarks, poi, tiger\_roads, giant\_polygon
- \* : tiger-nyc

OK Cancel

WICKET AJAX DEBUG

### Confirmer la suppression des entrepôts

## Une réponse efficace à la directive INSPIRE

### Couches

Dans GeoServer, le terme de couche se réfère aux données raster ou vecteur qui contiennent des objets géographiques. Les couches vecteurs sont analogues au *featureTypes* et les couches raster analogues au *coverages*. Les couches représentent chaque objet qui doivent être affichés sur la carte. Toutes les couches ont une source de données, appellé *Store*.

Dans la section couches, vous pouvez voir et éditer des couches existantes, ajouter (enregistrer) une nouvelle couche ou effacer (désenregistrer) une couche. Comme dans la table Vues précédente, la page de visualisation des couches affiche les dépendances appropriées, i.e. les couches dans le store dans l'espace de nom. La page Vue affiche également les status des couches et le SRS natif.

### Layers

Manage the layers being published by GeoServer

- [Add a new resource](#)
- [Remove selected resources](#)

Results 1 to 10 (out of 19 items)					
Type	Workspace	Store	Layer Name	Enabled?	Native SRS
	nurc	arcGridSample	Arc_Sample		EPSG:4326
	nurc	img_sample2	Pk50095		EPSG:32633
	nurc	mosaic	mosaic		EPSG:4326
	nurc	worldImageSample	Img_Sample		EPSG:4326
	sf	sf	archsites		EPSG:26713
	sf	sf	bugsites		EPSG:26713
	sf	sf	restricted		EPSG:26713
	sf	sf	roads		EPSG:26713
	sf	sf	streams		EPSG:26713
	sf	sfdem	sfdem		EPSG:26713

Vue de couches

### Types de couches

Les couches sont organisées en deux types de données, raster et vecteur. La différence entre ces deux formats est la manière dont il stocke l'information spatiale. Les types vecteurs stocke l'information sous forme de type d'objet en tant que chemin mathématique -- un point comme une unique coordonnées x,y, des lignes comme une série de coordonnées x,y et des polygones comme une série de coordonnées x,y qui démarre et se termine au même endroit. Les données au format raster est une représentation basée sur des cellules d'objet surfacique de la terre. Chaque cellule possède une valeur distincte, et toutes les cellules avec une valeur identique représentent un objet spécifique.

Champ	Description
	raster (grille)
	vecteur (feature)

### Éditez les données d'une couche

Une réponse efficace à la directive INSPIRE

Cliquer le nom de la couche ouvre un panneau de configuration de couche. L'onglet *Données*, activé par défaut, vous permet de définir et changer les paramètres des données pour une couche.

## nurc:Arc\_Sample

Configure the resource and publishing information for the current layer

**Data**    **Publishing**

**Basic Resource Info**

**Name**  
Arc\_Sample

**Title**  
A sample ArcGrid file

**Abstract**

**Keywords**

**Current Keywords**

WCS  
arcGridSample  
arcGridSample\_Coverage

**Add**

Vues des données des couches

### **Info basique**

La section qui débute -- Information basique des ressources, mots clés et lien de métadonnées sont analogues à la section *Service Metadata* pour le WCS, WFS et WMS. Ces sections fournissent des "données sur les données," précisément des informations textuelles qui permettent de travailler plus facilement sur les couches.

**Nom :** Identifiant utilisé pour référencer la couche dans les requêtes WMS.

**Titre :** Une description lisible pour identifier brièvement la couche au client (requis)

**Résumé :** Fournit une description plus longue avec plus d'information sur la couche.

## Une réponse efficace à la directive INSPIRE

**Mots-clés** : Liste de mots courts associés à la couche pour aider dans la recherche du catalogue.

**Lien de métadonnées** : Permet de lier des documents externes qui décrivent la couche de données. Pour le moment seulement deux types de formats standards sont valides : TC211 et FGDC. TC211 fait référence à la structure de métadonnées établies par le [Comité Technique ISO pour l'information Géographique/Géomatique](#) (ISO/TC 211) tandis que FGDC fait référence à ceux définie par le [Comité des données géographiques fédérales](#) (FGDC) des États-Unis.

Ajouter un lien de métadonnées au format FGDC

## Systems de référence des coordonnées

Un Système de Coordonnées de Référence (CRS) définit comment les données spatiales géoréférencées sont représentées par rapport à leur localisation réelle sur la surface de la terre. CRS font partie du modèle générale appelé Systèmes de Références Spatiales (SRS) qui inclut le référencement par des identifiant géographique et de coordonnées. Géoserver a besoin de savoir le Système de Référence de Coordonnées de vos données. Cette information est utilisé pour calculer la boîte englobante en latitude/longitude et projeter pendant les requêtes WMS et WFS.

Ajouter un CRS à une couche

**SRS native** : fait référence à la projection de la couche. Cliquer sur le lien projection affichera une description du SRS.

**SRS déclaré** : fait référence à ce que GeoServer envoie aux clients.

**SRS pris en charge** : détermine comment GeoServer doit prendre en charge la projection quand les deux SRS diffèrent.

## Boîtes englobantes

La boîte englobante détermine l'étendue d'une couche. La *boîte englobante native* sont les limites des données projetées dans la SRS native. Vous pouvez générer ces limites en cliquant le bouton *Calculer à partir des données*. Le bouton *Boîte englobante Lat/Long* calcule les limites basées sur le standard lat/long. Ces limites peuvent être générées en cliquant sur le bouton *Calculer les limites natives*.

Boîte englobante sf:archsites

## Paramètres de coverage (Raster)

Les paramètres optionnels des coverage sont possible pour certaines dtypes de données raster. Les formats worldImage demandent un domaine valide de coordonnées grilles en deux dimensions connu comme *ReadGridGeometry2D*. Pour ImageMosaic, vous pouvez utiliser *InputImageThresholdValue*, *InputTransparentColor*, et *OutputTransparentColor* pour contrôler le rendu de la mosaïque en terme de seuil et de transparence.

Une réponse efficace à la directive INSPIRE

### Détail des Feature Type (Vecteur)

À la place des paramètres de couvertures, les couches vecteurs ont une liste de *Détails de Feature Type*. Cela inclut Propriété et Type de source de données. Par exemple, la couche sf:archsites affichée ci-dessous inclut une géométrie, the\_geom de type point.

Feature Type Details			
Property	Type	Nillable	Min/Max Occurrences
the_geom	Point	true	0/1
cat	Long	true	0/1
str1	String	true	0/1

### Détails du Feature Types pour sf:archsites

Le *Nillable* fait référence à la nécessité d'avoir une valeur pour une propriété ou si elle peut être null (vide). Alors que *Occurrences Min/Max* fait référence au nombre de valeurs que le champ est autorisé à avoir. Pour le moment *Nillable* et *Occurrences Min/Max* sont définis à true et 0/1 mais peut être étendue avec un travail supplémentaire sur les features traitées.

### Éditez les informations de publication

L'onglet publication permet la configuration des paramètres HTTP et WCS.

## nurc:Arc\_Sample

Configure the resource and publishing information for the current layer

**Data**   **Publishing**

---

### Basic Settings

Name  
Arc\_Sample

Enabled

---

### HTTP Settings

Response Cache Headers

Cache Time (seconds)  
[ ]

---

### WCS Settings

#### Request SRS

Current Request SRS List  
EPSG:4326  Delete Selected

New Request SRS  
[ ]

---

#### Response SRS

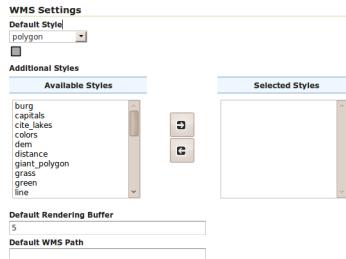
Current Response SRS List  
EPSG:4326  Delete Selected

New Request SRS  
[ ]

### Édition des données publiées

**Paramétrage HTTP** : paramètres de cache qui s'applique aux réponses HTTP à partir des requêtes clients. Si *En-tête du cache de la réponse* est coché, GeoServer n'interrogera pas la même tuile deux fois dans un temps défini dans *Durée du cache*. Une heure mesurée en seconde (i.e., 3600), est la valeur par défaut de *Cache Time*.

**Paramétrages WMS** : définie les paramètres de publication spécifique au WMS.



- *Style par défaut* : le style qui sera utilisé quand le client ne définit pas un style nommé dans les requêtes GetMap.
- *Styles supplémentaires* : autres styles qui peuvent être associés à cette couche. Certains clients (et celui de prévisualisation de GeoServer) présenteront ceux-ci comme styles alternatifs pour cette couche à l'utilisateur final.
- *Buffer de rendu par défaut* (disponible depuis la version 2.0.3) : la valeur par défaut du paramètre fourni par GetMap/GetFeatureInfo buffer. Voir `wms_vendor_parameters` pour plus de détails.
- *Chemin par défaut du WMS* : la localisation de la couche dans l'arbre de couches des capacités du WMS. Utile pour construire des groupes de couches non opaques.

**Attribution WMS** : définie les informations de publication sur les fournisseurs de données.

#### Attribution WMS

- *Texte d'attribution* : Texte lisible décrivant le fournisseur de données. Cela peut être utilisé comme texte pour un lien HTML vers le site web du fournisseur de données.
- *Lien d'attribution* : une URL vers le site web du fournisseur de données.
- *URL du logo* : une URL vers une image qui sert de logo pour le fournisseur de données.
- *Content Type, largeur et hauteur du logo* : ces champs fournissent des informations sur l'image du logo que les clients peuvent utiliser pour aider à la mise en page. GeoServer auto-déetectera ces valeurs si vous cliquez sur le lien *Auto-déTECTer le type et la taille de l'image* en bas de la section.

Le texte, lien et URL sont chacun affichés dans le document Capabilities WMS s'ils sont définis ; certains clients WMS les afficheront pour permettre aux utilisateurs de savoir quel fournisseur fournit un jeu de données particulier. Si vous oubliez certains champs, ceux qui sont définis seront publiés et ceux qui ne le sont pas seront omis du document Capabilities.

**Paramétrages WFS** : pour la couche, définie un nombre maximal de features qu'une opération GetFeature WFS peut générer, sans liaison avec le nombre de requêtes.

Une réponse efficace à la directive INSPIRE

**Paramètrages WCS** : fournie une liste de SRS vers lesquelles une couche peut être convertie. *Nouvelle requête SRS* vous permet d'ajouter un SRS à cette liste.

**Méthodes d'interpolation** : définit le processus de rendu du raster.

**Formats** : liste quels formats de sortie sont gérés par une couche.

**Titre par défaut** : assigne un style à une couche. Des styles supplémentaires sont publiés avec la couche dans le document capabilities.

**Geosearch** : lorsqu'il est activé, permet au crawler de Google Geo search d'indexer cette couche particulière. Voyez [Qu'est ce qu'un Sitemap géographique ?](#) pour plus d'information.

**Paramètrages du format KML** : Permet de limiter les entités basé sur certains critères. Choisissez quelles entités doivent s'afficher d'une manière plus visible que les autres avec *guilabel:Attribut par défaut de régionalisation*. Il y a quatre types de *Méthodes de régionalisation* :

- *external-sorting*: crée une base de données auxiliaire au sein de GeoServer. Cela prend un peu de temps supplémentaire pour construire l'index lors de la première requête.
- *geometry*: classe en externe par longueur (si ce sont des lignes) ou par surface (si ce sont des polygones).
- *native-sorting*: utilise l'algorithme de classement par défaut du backend dans lequel sont stockées les données. Cela est plus rapide que l'*external-sorting*, mais fonctionnera seulement avec les datastores PostGIS.
- *random*: utilise l'ordre existant des données et ne les classera pas.

### Ajouter un supprimer une couche

En haut à gauche de la page de la liste des couches se trouve deux boutons pour l'ajout et la suppression des couches. Le bouton "plus" en vert permet d'ajouter une nouvelle couche, ici nommée comme ressource. Le bouton "moins" en rouge permet de supprimer des couches sélectionnées.

## Layers

Manage the layers being published by GeoServer

 Add a new resource

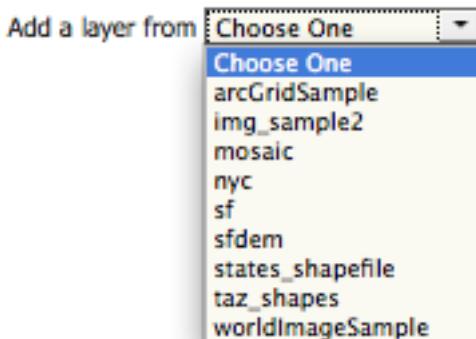
 Remove selected resources

Bouton pour ajouter ou supprimer une couche

Cliquer sur le bouton *Ajouter une nouvelle ressource* affiche une fenêtre *Choix de la nouvelle couche*. La liste déroulante affiche tous les stores activés. À partir de ce menu sélectionnez le store dans lequel se trouve la couche qui doit être activée.

Une réponse efficace à la directive INSPIRE

## New Layer chooser



### Liste de tous les stores activés

Après la sélection du store, une table de vue des couches existantes dans le store sélectionné sera affichée. Dans cet exemple, giant\_polygon, poi, poly\_landmarks et tiger\_roads sont toutes les couches dans le store NYC.

## New Layer chooser

A screenshot of the "New Layer chooser" interface for the "nyc" store. At the top left, there is a text input field labeled "Add a layer from" with "nyc" selected. Below it, a message says "Here is a list of resources contained in the 'nyc' store. Click on the layer you wish to configure". A search bar with a magnifying glass icon is also present. The main area shows a table with two columns: "Layer with namespace and prefix" and "Published". The table contains four rows: "giant\_polygon" (published), "poi" (published), "poly\_landmarks" (published), and "tiger\_roads" (published). At the bottom, there are navigation buttons for the list: &lt;&lt;, &lt;, &gt;, &gt;&gt; and a message "Results 0 to 0 (out of 0 items)".

### Vues de toutes les couches

Après la sélection d'un nom de couche, vous êtes redirigés vers une page d'édition de couche, [Éditez les données d'une couche](#)

Pour supprimer une couche, cliquez sur la case à cocher présente sur la gauche de chaque couche. Comme indiqué plus bas, plusieurs couches peuvent être cochées pour suppression sur une seule page. Cependant la sélection pour supprimer ne persistera pas d'une page vers une autre.

## Une réponse efficace à la directive INSPIRE

### Layers

Manage the layers being published by GeoServer

- [Add a new resource](#)
- [Remove selected resources](#)

Results 1 to 10 (out of 18 items)						Search
Type	Workspace	Store	Layer Name	Enabled?	Native SRS	
	nurc	img_sample2	Pk50095		EPSG:32633	
	nurc	mosaic	mosaic		EPSG:4326	
	nurc	worldImageSample	Img_Sample		EPSG:4326	
	sf	sf	archsites		EPSG:26713	
	sf	sf	bugsites		EPSG:26713	
	sf	sf	restricted		EPSG:26713	
	sf	sf	roads		EPSG:26713	
	sf	sf	streams		EPSG:26713	
	sf	sfdem	sfdem		EPSG:26713	
	tiger	nyc	giant_polygon		EPSG:4326	

Couches nurc:Img\_Sample, sf:restricted, sf:streams sélectionnées pour suppression

Toutes les couches peuvent être sélectionnées pour suppression en activant la case à cocher dans la ligne d'en-tête.

Results 1 to 1		
<input checked="" type="checkbox"/>	Type	Workspace
<input checked="" type="checkbox"/>		nurc
<input checked="" type="checkbox"/>		nurc
<input checked="" type="checkbox"/>		nurc
<input checked="" type="checkbox"/>		sf
<input checked="" type="checkbox"/>		tiger

Toutes les couches sélectionnées pour suppression

## Une réponse efficace à la directive INSPIRE

Une fois que les couches ont été sélectionné, le lien *Retirer les ressources sélectionnées* est activé. En cliquant sur le lien, une confirmation vous sera demandée. La sélection de *OK* supprime la couche.

### **Groupes de couches**

Un groupe de couches est un groupe de couches qui peuvent être référencé par un seul nom. Cela permet des requêtes WMS plus simples, puisqu'il n'est nécessaire de référencer une seule couche dans la requête par opposition aux multiples couches individuelles. Les groupes de couches agissent simplement comme des couches standards au sein des services WMS.

#### **Layer Groups**

Define and manage layer groupings

 Add new layer group

 Remove selected layer group(s)

<< < 1 > >> Results 1 to 3 (out of 3 items)

 Search

Layer Group

spearfish

tasmania

tiger-ny

<< < 1 > >> Results 1 to 3 (out of 3 items)

Page des groupes de couches

### **Éditer un groupe de couches**

Pour afficher la page d'édition d'un groupe de couches, cliquez sur un nom de groupe de couche. Les champs initiaux permettent la configuration du nom, des limites et la projection du groupe de couches. Pour définir automatiquement la bounding box, sélectionnez le bouton *Générez les limites*, sinon mettez y les limites personnalisées. Pour sélectionner une projection appropriée cliquez sur le bouton *Trouver*.

#### **Note**

Un groupe de couche peut consister de couches avec des projections et des limites différentes. GeoServer va automatiquement reprojeter toutes les couches du groupe de couches.

En bas de la page il y a un tableau de couches contenu dans le groupe de couches actuel. Quand un groupe de couches est réalisé, les couches sont rendues dans l'ordre donné, la couche en bas de la liste sera rendue en dernière et donc sera affichée par dessus les autres.

## Une réponse efficace à la directive INSPIRE

### Layer group

Edit the contents of a layer groups

Name  
spearfish

Bounds

Min X	Min Y	Max X	Max Y
589,425.934	4,913,959.225	609,518.672	4,928,082.95

EPSG:26713      Find...      EPSG:NAD27 / UTM zone 13N...

Generate Bounds

Add Layer...

Layers

Layer	Style	Remove	Position
sfdem	dem	✖	↓
streams	simple_streams	✖	↑ ↓
roads	simple_roads	✖	↑ ↓
restricted	restricted	✖	↑ ↓
archsites	point	✖	↑ ↓
bugsites	capitals	✖	↑

<< < 1 > >> Results 1 to 6 (out of 6 items)

### Page d'édition des groupes de couches

La colonne *Style* affiche le style associé avec chaque couches. Pour changer le style associé à la couche, cliquez sur le lien du style. Une liste de styles activés sera affichée. Cliquez sur le nom du style pour l'assigner comme style de la couche.

Layer group

Edit the contents of a layer groups

Name  
spearfish

Bounds

Min X	Min Y	Max X	Max Y
589,425.934	4,913,959.225	609,518.672	4,928,082.95

EPSG:26713      Find...      EPSG:NAD...

Generate Bounds

Add Layer...

Layers

Layer	Style	Remove	Position
sfdem	dem	✖	↓
streams	simple_streams	✖	↑ ↓
roads	simple_roads	✖	↑ ↓
restricted	restricted	✖	↑ ↓
archsites	point	✖	↑ ↓
bugsites	capitals	✖	↑

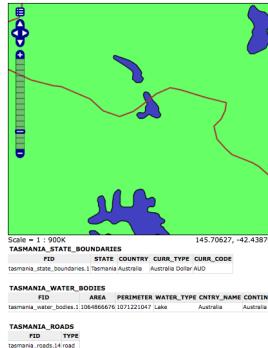
<< < 1 > >> Results 1 to 22 (out of 22 items)

### Édition du style pour une couche dans le groupe de couche

## Une réponse efficace à la directive INSPIRE

Pour supprimer une couche d'un groupe de couches, sélectionnez le bouton de la couche dans la colonne *Supprimer*. On vous demandera alors de confirmer ou d'annuler la suppression.

Vous pouvez voir le groupe de couches dans la section *Prévisualisation de couches* de la console d'admin web.



Prévisualisation avec Openlayers du groupe de couches "tasmania"

Une couche peut être positionnée plus haut ou plus bas dans la liste en cliquant sur les flèches vertes 'haut' et 'basse' respectivement.

Une couche peut être ajoutée à la liste en pressant le bouton *Ajouter une couche...* en haut du tableau de couches. À partir de la liste de couches résultantes, sélectionnez la couche à ajouter en cliquant sur le nom de la couche. Cette dernière sera ajoutée en bas de la liste des couches.

### Layer group

Edit the contents of a layer groups

Name: tasmania

Bounds:

Min X: 143.835	Min Y: -43.648	Max X: 148.479	Max Y: -35.100
EPSG:4326	Find...	EPSG:WGS84	

Generate Bounds

+ Add Layer...

Layers:

Layer
tasmania_water_bodies
tasmania_state_boundaries
tasmania_roads
tasmania_cities
Img_Sample

Results 1 to 4 (out of 4)

<< < > >> Results 1 to 19 (out of 19 items)

name	store	workspace
giant_polygon	nyc	tiger
poi	nyc	tiger
poly_landmarks	nyc	tiger
tiger_roads	nyc	tiger
archsites	sf	sf
bugsites	sf	sf
restricted	sf	sf
roads	sf	sf
streams	sf	sf
states	states_shapefile	topp
tasmania_cities	taz_shapes	topp
tasmania_roads	taz_shapes	topp

<< < > >> Results 1 to 4 (out of 4)

Dialogue pour ajouter une couche au groupe de couches

## Ajouter un groupe de couches

Une réponse efficace à la directive INSPIRE

Les boutons pour ajouter et supprimer un groupe de couche sont situés en haut de la page *Groupes de couches*.

## Layer Groups

Define and manage layer groupings

 Add new layer group

 Remove selected layer group(s)

*Boutons pour ajouter et supprimer un groupe de couches*

Pour ajouter un nouveau groupe de couches, sélectionnez le bouton "Ajouter un groupe de couches". On vous demandera le nom du groupe de couches.

## New Layer Group

Add a new layer grouping

Name

*Dialogue de nouveau groupe de couches*

Lorsque c'est terminé, cliquez sur *Soumettre*. Vous serez alors redirigé vers une page de configuration de groupe de couche vide. Commencez en ajoutant des couches en cliquant sur le bouton *Ajouter une couche...* (décris dans la section précédente). Un fois que les couches sont positionnées correctement, pressez *Générez les limites* pour générer automatiquement la projection et l'étendue de la couche. Pressez *Sauver* pour sauver le nouveau groupe de couche.

## Layer group

Edit the contents of a layer groups

Name

Bounds

Min X	Min Y	Max X	Max Y
-------	-------	-------	-------

...

 Add Layer...

Layers

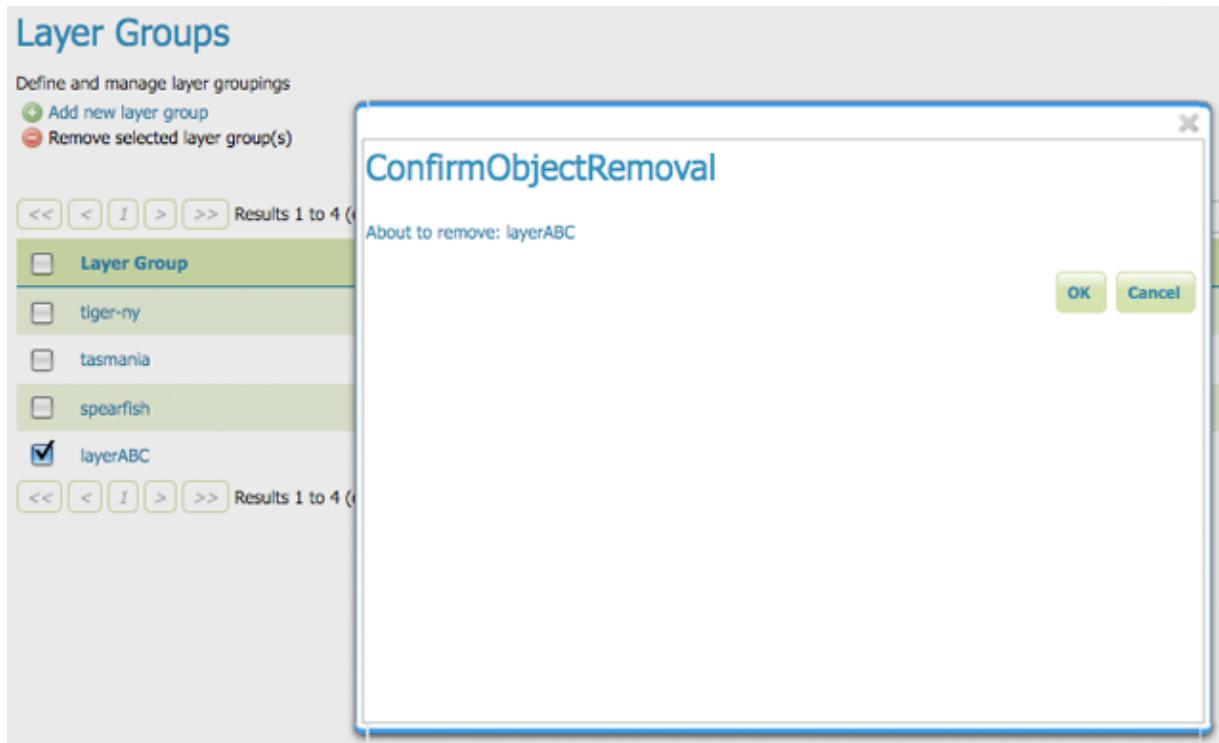
Layer	Style	Remove	Position
<< < > >>	Results 0 to 0 (out of 0 items)		

*Page de configuration d'un nouveau groupe de couches*

Une réponse efficace à la directive INSPIRE

### **Supprimer un groupe de couches**

Dans le but de supprimer un groupe de couches, cliquez sur la case à cocher près du groupe de couche. Plusieurs groupes de couches peuvent être sélectionnés pour la suppression. Cliquez sur le lien *supprimer le(s) groupe(s) de couches sélectionné(s)*. On vous demandera de confirmer ou d'annuler la suppression. Sélectionner OK supprime le groupe de couches.



*Supprimer un groupe de couches*

### **Styles**

Les styles sont la manière pour déterminer le rendu des données spatiales. Les styles pour GeoServer sont écrit en Styled Layer Descriptor (SLD), un sous-ensemble de XML. Lisez la section [Symbolisation](#) pour plus d'information sur comment utiliser les styles.

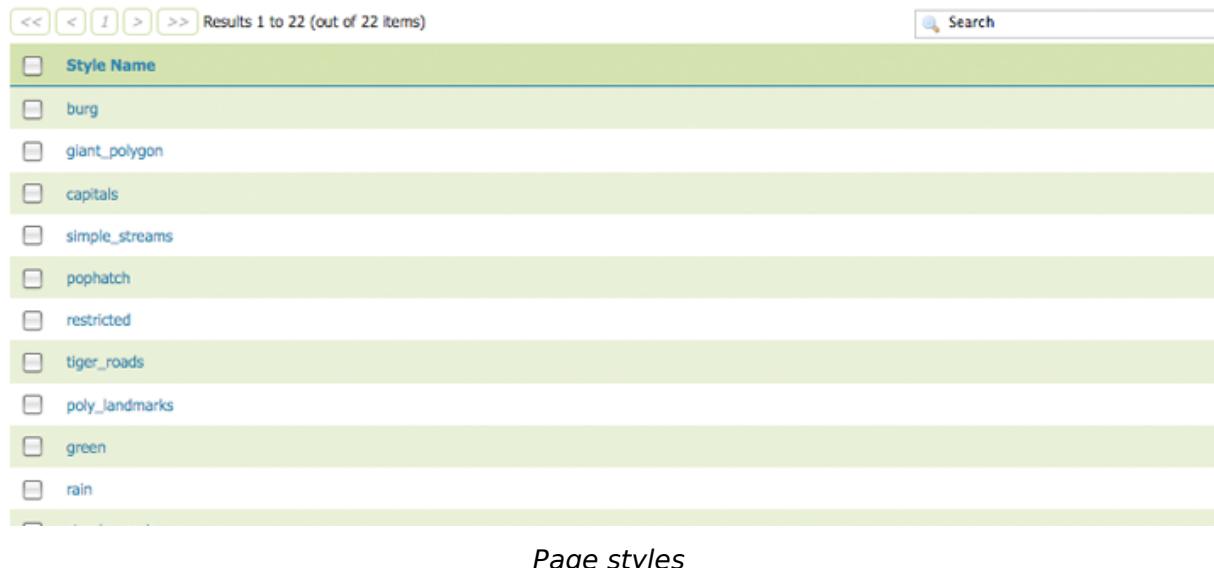
Sur cette page, vous pouvez enregistrer ou créer un nouveau style, éditer ou supprimer un style existant.

## Une réponse efficace à la directive INSPIRE

### Styles

Manage the Styles published by GeoServer

-  Add a new style
-  Removed selected style(s)



The screenshot shows a list of styles managed by GeoServer. At the top, there are navigation buttons (<<, <, I, >, >>) and a search bar. Below this, a message says "Results 1 to 22 (out of 22 items)". The list itself has a header "Style Name" with a checkbox column. Each item in the list contains a checkbox followed by the style name: burg, giant\_polygon, capitals, simple\_streams, pophatch, restricted, tiger\_roads, poly\_landmarks, green, and rain. The entire list is contained within a light green horizontal bar.

Page styles

### Éditer des styles

La page *Éditeur de style* présente des options pour configurer le nom et le code d'un style. Les noms des SLD sont définie en haut dans le champs nom. Entrer ou coller le code SLD peut être réalisé en un ou deux modes. Le premier mode est un éditeur riche inclu dans un **EditArea**. Le second mode est un éditeur de texte non formaté. Cliquez sur le bouton *éditeur* pour passer d'un mode à l'autre.

## Style Editor

Edit the current Styled Layer Description style. The editor can provide syntax highlight and be brought to full screen. Click on the "validate" |

### Name

giant\_polygon

### SLD Text

The screenshot shows a rich text editor interface for editing SLD (Styled Layer Descriptor) XML. The XML code is syntax-highlighted, showing tags in blue and values in black. The code defines a named layer named 'area landmarks' with a title 'Border-less gray fill'. It uses a light gray polygon fill without a border. The editor has a toolbar at the top with icons for bold, italic, underline, etc., and a font size dropdown set to 8pt. Below the toolbar is the XML code. At the bottom, there's a status bar showing 'Position: Ln 1, Ch 1' and 'Total: Ln 27, Ch 1002'. Below the status bar are three buttons: 'Validate' (green), 'Submit' (green), and 'Cancel' (gray). A 'Toggle editor' link is also present.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/def/crs/OGC/1.3/CRS_OGC" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sld http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd">
  <NamedLayer>
    <Name>area landmarks</Name>
    <UserStyle>
      <Title>Border-less gray fill</Title>
      <Abstract>Light gray polygon fill without a border</Abstract>
      <FeatureTypeStyle>
        <Rule>
          <PolygonSymbolizer>
            <Fill>
              <CssParameter name="fill">
                <ogc:Literal>#DDDDDD</ogc:Literal>
              </CssParameter>
              <CssParameter name="fill-opacity">
                <ogc:Literal>1.0</ogc:Literal>
              </CssParameter>
            </Fill>
          </PolygonSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

Éditeur de texte riche

Une réponse efficace à la directive INSPIRE

## Style Editor

Edit the current Styled Layer Description style. The editor can provide syntax highlight and be brought to full the SLD schema.

Name	<input type="text" value="giant_polygon"/>
SLD Text	<pre>     &lt;?xml version="1.0" encoding="ISO-8859-1"?&gt;     &lt;StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/def/crs/OGC/1.3/CRS_OGC"&gt;         &lt;NamedLayer&gt;             &lt;Name&gt;area landmarks&lt;/Name&gt;             &lt;UserStyle&gt;                 &lt;Title&gt;Border-less gray fill&lt;/Title&gt;                 &lt;Abstract&gt;Light gray polygon fill without a border&lt;/Abstract&gt;                 &lt;FeatureTypeStyle&gt;                     &lt;Rule&gt;                         &lt;PolygonSymbolizer&gt;                             &lt;Fill&gt;                                 &lt;CssParameter name="fill"&gt;                                     &lt;ogc:Literal&gt;#DDDDDD&lt;/ogc:Literal&gt;                                 &lt;/CssParameter&gt;                                 &lt;CssParameter name="fill-opacity"&gt;                                     &lt;ogc:Literal&gt;1.0&lt;/ogc:Literal&gt;                                 &lt;/CssParameter&gt;                             &lt;/Fill&gt;                         &lt;/PolygonSymbolizer&gt;                     &lt;/Rule&gt;                 &lt;/FeatureTypeStyle&gt;             &lt;/UserStyle&gt;         &lt;/NamedLayer&gt;     </pre>
	<input type="button" value="Validate"/> <input type="button" value="Submit"/> <input type="button" value="Cancel"/>

## *Éditeur de texte simple*

L'éditeur riche est conçu pour le formatteur de texte, la recherche et le remplacement, la numérotation de ligne et la surcharge syntaxique en temps réel. Vous pouvez aussi passer en mode plein écran pour une zone d'édition plus grande.

Bouton	Description
	recherche
	aller à la ligne
	mode plein écran
	annuler
	refaire
	alterner la surcharge de la syntaxe en on/off

Une réponse efficace à la directive INSPIRE

	reset la surcharge de la syntaxe (lorsqu'elle n'est plus synchroniser)
	à propos

Pour confirmer que le code SLD est pleinement conforme avec le schéma SLD, pressez le bouton *Valider*. Une boîte de messafe confirmara si le style possède des erreurs de validation.

### Note

GeoServer pourra parfois utiliser un style qui ne passe pas la validation, mais cela n'est pas recommandé.

No validation errors.

## Style Editor

Aucune erreur de validation

org.xml.sax.SAXParseException: The element type "NamedLayer" must be terminated by the matching end-tag "</NamedLayer>".

## Style Editor

Message d'erreur de validation

### Ajouter un style

Les boutons pour ajouter et supprimer un style peuvent être trouvé en haut de la page *Styles*.

## Styles

Manage the Styles published by GeoServer

- Add a new style
- Removed selected style(s)

Boutons pour ajouter et supprimer un style

Pour ajouter un nouveau groupe de couche, sélectionnez le bouton *Ajouter un nouveau style*. Vous serez redirigé vers une page d'édition. Entrez un nom pour le style. La page d'édition propose deux options pour soumettre un SLD. Vous pouvez coller directement le SLD dans l'éditeur ou vous pouvez télécharger un fichier local qui contient le SLD.

SLD file

Télécharger un fichier SLD à partir de votre ordinateur

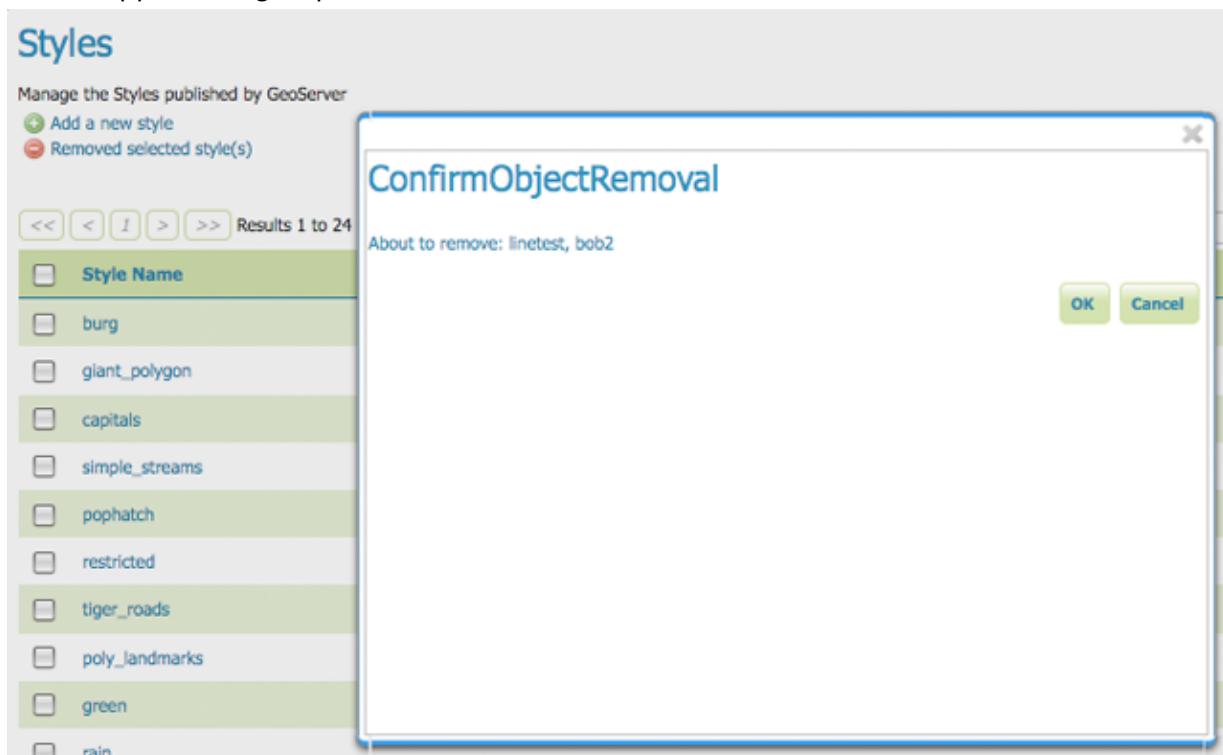
Une fois que le style est soumis, vous serez redirigé vers la page principale des *Styles* où le style sera listé.

### Supprimer un style

Pour supprimer un style, cliquez sur la case à cocher près du style. Plusieurs groupes de couches peut être cochés pour suppression multiple. Cliquez sur le lien *Supprimer les styles*

Une réponse efficace à la directive INSPIRE

sélectionnés en haut de la page. Vous devrez confirmer la suppression ou annuler. Cliquer sur OK supprime le groupe de couche.



*Message de confirmation pour la suppression d'un style*

## Démos

Cette page contient des liens utiles vers différentes pages d'information sur GeoServer et ses fonctionnalités. Vous n'avez pas besoin d'être identifié dans GeoServer pour accéder à cette page.

## GeoServer Demos

Collection of GeoServer demo applications and tools

- [Demo requests](#) Example requests for GeoServer (using the TestServlet).
- [SRS List](#) List of all SRS known to GeoServer

*Page de démos*

## Requêtes de démo

Cette page propose des exemples de requêtes WMS, WFS et WCS pour GeoServer que vous pouvez utiliser, examiner et modifier. Sélectionnez une requête à partir de la liste déroulante.

Une réponse efficace à la directive INSPIRE

## Demo requests

Example requests for GeoServer (using the TestServlet). Select a request from the drop down list, and then hit 'Change'. This will display the request to send the request to GeoServer.

Request	<input type="button" value="Choose One"/>
URL	<input type="text" value="WCS_describeCoverage.xml"/> WCS_getCapabilities.xml WCS_getCoverage.xml WFS_describeFeatureType-1.0.xml WFS_describeFeatureType-1.1.xml WFS_getCapabilities-1.0.xml WFS_getCapabilities-1.1.xml WFS_getFeature-1.0.xml WFS_getFeature-1.1.xml WFS_getFeature880X-1.0.xml WFS_getFeature880X-1.1.xml WFS_getFeature880X.url WFS_getFeatureBetween-1.0.xml WFS_getFeatureBetween-1.1.xml WFS_getFeatureBetween.url WFS_getFeatureBetweenCQL.url WFS_getFeatureFid.url WFS_getFeatureFidFilter.url WFS_getFeatureIntersects-1.0.xml
Body	
User Name	<input type="text"/>
	<input type="text" value="Password"/>
<input type="button" value="Submit"/>	

### Selection des requêtes de démo

Les requêtes Web Feature Service (wfs) et les Web Coverage Service (wcs) afficheront l'URL de la requête et le corps XML. Les requêtes des Web Map Service (wms) afficheront seulement l'URL de la requête.

## Une réponse efficace à la directive INSPIRE

### Demo requests

Example requests for GeoServer (using the TestServlet). Select a request from the drop down list, and then hit 'Change'. This will display the request url (and body if an xml request). Hit submit to send the request to GeoServer.

Request

URL

Body

```
<!-- A sample describe request. The schema is generated automatically by -->
<!-- GeoServer. You can modify the schema with the web interface to hide -->
<!-- and/or require certain attributes. -->
<!--
    If you change the "<TypeName>" tag below to the name of another
    dataset, you can see the GML Schema for that layer.
    This will have all the column names and types.
    The getCapabilities demo will tell you the names of all the layers!
-->
<DescribeFeatureType
    version="1.1.0"
    service="WFS"
    xmlns="http://www.opengis.net/wfs"
    xmlns:topp="http://www.openplans.org/topp"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
    <TypeName>topp:states</TypeName>
```

User Name  Password

### Requête DescribeFeatureType d'exemple pour le WFS 1.1

Cliquez sur Soumettre pour envoyer la requête à GeoServer. Pour les requêtes WFS et WCS, GeoServer générera automatiquement une réponse XML.

Demo requests

Example requests for GeoServer (using the TestServlet). Select a request from the drop down list, and then hit 'Change'. This will display the request url (and body if an xml request). Hit submit to send the request to GeoServer.

Request

URL

Body

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

- <xsd:schema elementFormDefault="qualified" targetNamespace="http://www.openplans.org/topp">
  - <xsd:import namespace="http://www.opengis.net/gml" schemaLocation="http://localhost:8090/geoserver_latest/schemas/gml/3.1.1/base/gml.xsd"/>
  - <xsd:complexType name="statesType">
    - <xsd:complexContent>
      - <xsd:extension base="gml:AbstractFeatureType">
        - <xsd:sequence>
          - <xsd:element maxOccurs="1" minOccurs="0" name="the_geom" nillable="true" type="gml:MultiSurfacePropertyType"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="STATE_NAME" nillable="true" type="xsd:string"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="STATE_FIPS" nillable="true" type="xsd:string"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="SUB_REGION" nillable="true" type="xsd:string"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="STATE_ABBR" nillable="true" type="xsd:string"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="LAND_KM" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="WATER_KM" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="PERSONS" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="FAMILIES" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="HOUSHOLD" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="MALE" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="FEMALE" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="WORKERS" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="DRVALONE" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="CARPOOL" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="PUBTRANS" nillable="true" type="xsd:double"/>
          - <xsd:element maxOccurs="1" minOccurs="0" name="EMPLOYED" nillable="true" type="xsd:double"/>
```

User Name

### Réponse XML à partir d'une requête DescribeFeatureType d'exemple pour du WFS 1.1

Soumettre un requête GetMap d'un WMS affichera une image basée sur les données géographiques fournies.

## Une réponse efficace à la directive INSPIRE

### Demo requests

Example requests for GeoServer (using the TestServlet). Select a request from the drop down list, and then hit 'Change'. This will display the request url (and body) and hit submit to send the request to GeoServer.

Request: WMS\_getMap\_OpenLayers.url  
URL: http://localhost:8090/geoserver/  
Body:  
User Name: admin  
Submit

OpenLayers map preview

Scale = 1 : 50M      -118.12500, 47.50000  
Click on the map to get feature info

Requête GetMap du service WMS par OpenLayers

Les requêtes GetFeatureInfo des services WMS récupère des informations d'une entité particulière d'une image cartographique.

### Demo requests

Example requests for GeoServer (using the TestServlet). Select a request from the drop down list, and then hit 'Change'. This will display the request url (and body) and hit submit to send the request to GeoServer.

Request: WMS\_featureInfo.url  
URL: http://localhost:8090/geoserver/  
Body:  
User Name: admin  
Submit

Results for FeatureType 'states':

```
the_geom = [GEOMETRY (MultiPolygon) with 153 points]
STATE_NAME = Arizona
STATE_FIPS = 04
SUB_REGION = Mtn
STATE_ABBR = AZ
LAND_KM = 294333.462
WATER_KM = 942.772
PERSONS = 3665228.0
FAMILIES = 940106.0
HOUSEHOLD = 1368843.0
MALE = 1810691.0
FEMALE = 1854537.0
WORKERS = 1358263.0
DRVALONE = 1178320.0
CARPOOL = 239083.0
PUBTRANS = 32856.0
EMPLOYED = 1603896.0
UNEMPLOY = 123902.0
SERVICE = 455896.0
MANUAL = 185109.0
P_MALE = 0.494
P_FEMALE = 0.506
SAMP_POP = 468178.0
```

Requêtes GetFeatureInfo des services WMS

## SRS

## Une réponse efficace à la directive INSPIRE

GeoServer gère nativement environ 4 à 100 Système de Référence Spatiale (SRS), appellé également **projection** et d'autres peuvent être ajoutés. Un système de Référence Spatiale définit un ellipsoïde, un datum en utilisant cet ellipsoïde et un système de coordonnées géocentrique, géographique ou projeté. Cette page liste toutes les informations SRS connues par GeoServer.

### SRS List

List of SRS known to GeoServer. You can choose the authority, filter based on the code and description, and gather details on each code

Code		Description
2000		Anguilla 1957 / British West Indies Grid
2001		Antigua 1943 / British West Indies Grid
2002		Dominica 1945 / British West Indies Grid
2003		Grenada 1953 / British West Indies Grid
2004		Montserrat 1958 / British West Indies Grid
2005		St. Kitts 1955 / British West Indies Grid
2006		St. Lucia 1955 / British West Indies Grid
2007		St. Vincent 45 / British West Indies Grid
2008		NAD27(CGQ77) / SCoPQ zone 2
2009		NAD27(CGQ77) / SCoPQ zone 3
2010		NAD27(CGQ77) / SCoPQ zone 4
2011		NAD27(CGQ77) / SCoPQ zone 5
2012		NAD27(CGQ77) / SCoPQ zone 6
2013		NAD27(CGQ77) / SCoPQ zone 7
2014		NAD27(CGQ77) / SCoPQ zone 8

*Liste de tous les Système de Référence Spatiale (SRS) connu de GeoServer*

La colonne *Code* fait référence à l'entier unique défini par l'auteur de ce système de référence spatiale. Chaque code est lié à une page de description plus détaillée que l'on peut visualiser en cliquant sur ce code.

## EPSG:2000

### Description

Anguilla 1957 / British West Indies Grid

### WKT

```
PROJCS["Anguilla 1957 / British West Indies Grid",
    GEOGCS["Anguilla 1957",
        DATUM["Anguilla 1957",
            SPHEROID["Clarke 1880 (RGS)", 6378249.145, AUTHORITY["EPSG","7012"]],
            AUTHORITY["EPSG","6600"]],
        PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]],
        UNIT["degree", 0.017453292519943295],
        AXIS["Geodetic longitude", EAST],
        AXIS["Geodetic latitude", NORTH],
        AUTHORITY["EPSG","4600"]],
    PROJECTION["Transverse Mercator", AUTHORITY["EPSG","9807"]],
    PARAMETER["central_meridian", -62.0],
    PARAMETER["latitude_of_origin", 0.0],
    PARAMETER["scale_factor", 0.9995],
    PARAMETER["false_easting", 400000.0],
    PARAMETER["false_northing", 0.0],
    UNIT["m", 1.0],
    AXIS["Easting", EAST],
    AXIS["Northing", NORTH],
    AUTHORITY["EPSG","2000"]]
```

### Area of validity

...

### Détails pour le SRS EPSG:2000

Le titre de chaque SRS est composé du nom de l'auteur et de l'identifiant entier unique (code) défini par l'auteur. Dans l'exemple ci-dessus, l'auteur est le [European Petroleum Survey Group](#) (EPSG) et le code est 2000. Les champs sont comme ceci :

*Description* : Une courte description du SRS.

*WKT* : Une chaîne de caractères décrivant le SRS. WKT signifie "Well Known Text."

*Zone de validité* : L'étendue du SRS.

### Prévisualisation de couches

Cette page propose des vues de couches dans différents formats de sortie. Notez qu'une couche doit être activée pour être prévisualisée.

## Une réponse efficace à la directive INSPIRE

### Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

Results 1 to 19 (out of 19 items)				
Type	Name	Title	Common Formats	All Formats
栅格	nurc:Arc_Sample	A sample ArcGrid file	OpenLayers KML	Select one
栅格	nurc:Pk50095	Pk50095 is a A raster file accompanied by a spatial data file	OpenLayers KML	Select one
栅格	nurc:mosaic	Sample PNG mosaic	OpenLayers KML	Select one
栅格	nurc:Img_Sample	North America sample imagery	OpenLayers KML	Select one
矢量	sf:archsites	Spearfish archeological sites	OpenLayers KML GML	Select one
矢量	sf:bugsites	Spearfish bug locations	OpenLayers KML GML	Select one
矢量	sf:restricted	Spearfish restricted areas	OpenLayers KML GML	Select one
矢量	sf:roads	Spearfish roads	OpenLayers KML GML	Select one
矢量	sf:streams	Spearfish streams	OpenLayers KML GML	Select one
栅格	sf:sfdem	sfdem is a Tagged Image File Format with Geographic information	OpenLayers KML	Select one
点	tiger:poi	Manhattan (NY) points of interest	OpenLayers KML GML	Select one

### Page de prévisualisation de couches

Chaque ligne de couche affiche un type, un nom un titre et les formats disponibles pour visualisation.

Type	Description
栅格	Couche raster (grille)
矢量	Couche vecteur (entité)
组	Groupe de couche

Le nom fait référence à l'espace de travail et au nom de la couche, alors que le titre fait référence à la courte description définie dans la partie [Éditez les données d'une couche](#). Dans l'exemple suivant, nurc fait référence à l'espace de travail, Arc\_Sample au nom de la couche et "A sample ArcGrid field" a été défini dans la partie édition des données de la couche.

Type	Name	Title	Common Formats	All Formats
栅格	nurc:Arc_Sample	A sample ArcGrid file	OpenLayers KML	Select one

### Une ligne de prévisualisation de couche unique

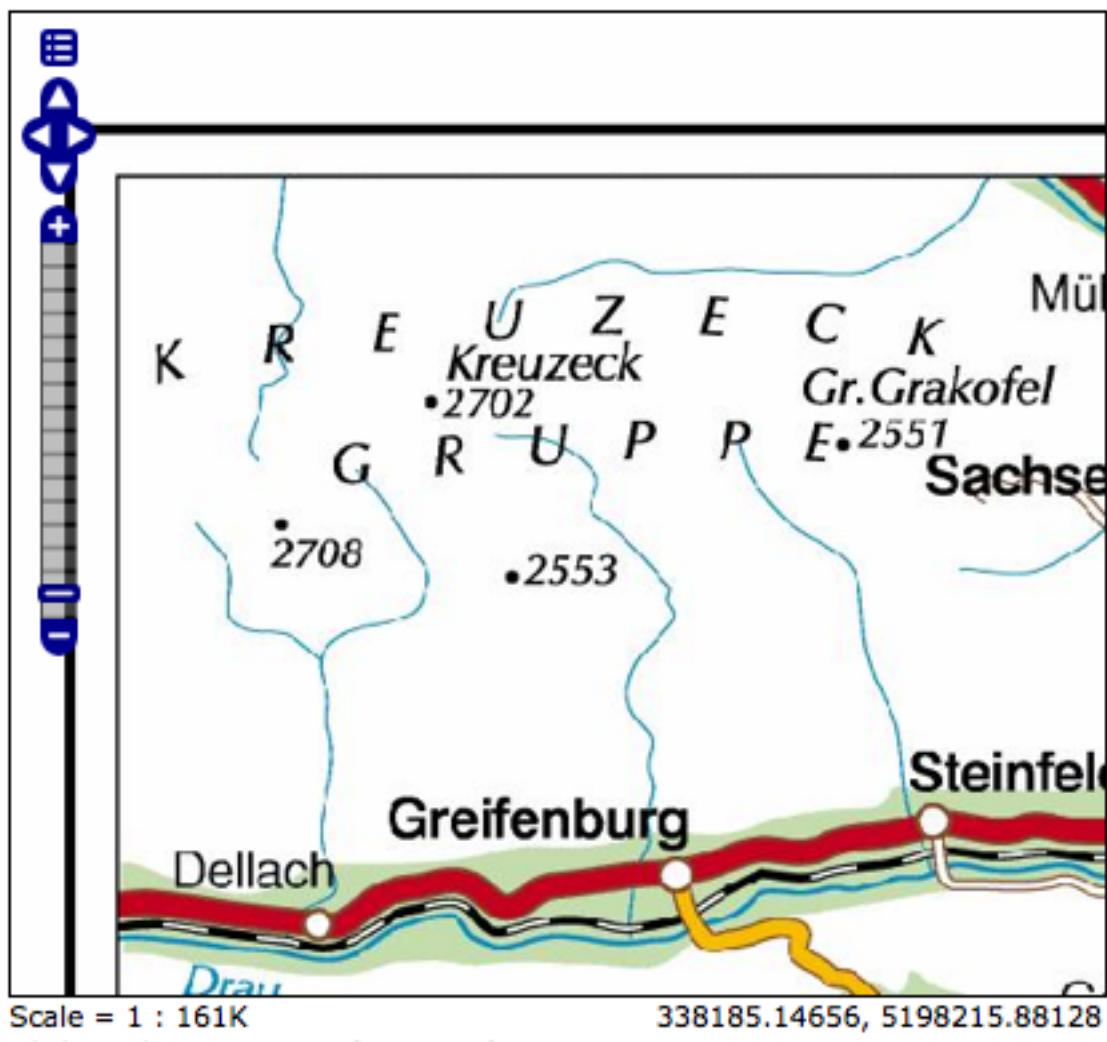
### Formats de sortie

La page de prévisualisation des couches gère différents formats de sortie pour des utilisations diverses ou du partage de données. Vous pouvez prévisualiser toutes les types de couches dans les formats OpenLayers et KML. Parallèlement en utilisant la liste déroulante "Tous les formats" vous pouvez voir tous les types de couches dans une demi douzaine de formats supplémentaires -- AtomPub, GIF, GeoRss, JPEG, KML (compressé), PDF, PNG, SVG, et TIFF. Seules les couches vecteurs proposent des prévisualisations de sortie WFS, dont le format GML et CSV, GML3, GeoJSON et Shapefile. Le tableau ci-dessous propose une brève description de tous les formats gérés, organisé en type de sortie : image, text ou données.

### **Sortie image**

Toutes les sorties d'images peuvent être réalisées à partir d'une requête GetMap WMS sur une données raster, vecteur ou couverture. Les services WMS sont des méthodes qui permettent l'affichage visuel de données spatiales sans nécessairement fournir un accès aux entités qui composent les données.

<b>Format</b>	<b>Description</b>
KML	KML (Keyhole Markup Language) est un schéma basé sur le langage XML pour modéliser des données géographiques afin de naviguer sur la terre comme Google Earth ou Google Maps. Le format KML utilise une structure basée sur des balises et des attributs imbriqués. Pour GeoServer, les fichiers KML sont distribués sous la forme de KMZ, qui sont des fichiers KML compressé.
JPEG	Sortie WMS en format raster. Le format JPEG est un format de fichier compressé avec perte de qualité dû à la compression. Il est conseillé pour les photos et pas pour la reproduction de données exactes.
GIF	Sortie WMS en format raster. Le format GIF (Graphics Interchange Format) est un format image bitmap utile pour les images en lignes de bordures nettes avec un nombre limité de couleur. Cela permet d'optimiser la compression sans perte du format qui favorise les zones étendues de même couleur avec des limites nettes (contrairement au format JPEG, qui favorise les images avec des gradients de couleurs). Le format GIF est limité à une palette de 8 bit ou 256 couleurs.
SVG	Sortie WMS de format vecteur. SVG (Scalable Vector Graphics) est un langage pour modéliser les graphiques en deux dimensions en XML. Il diffère des formats GIG et JPEG dans la mesure où il utilise des objets graphiques plutôt que des points individuels.
TIFF	Sortie WMS en format raster. Le format TIFF (Tagged Image File Format) est un format flexible, adaptable pour prendre en charge des données multiples dans un seul fichier. GeoTIFF contient des données géographiques incluses sous forme de balises dans le fichier TIFF.
PNG	Sortie WMS en format raster. Le format de fichier PNG (Portable Network Graphics) a été créé comme le successeur libre et open source au GIF. Le format de fichier PNG gère le truecolor (16 million de couleurs) tandis que le GIF gère seulement 256 couleurs. Le fichier PNG est performant lorsque l'image a des zones larges et de couleurs uniformes.
OpenLayers	Sortie de la requête GetMap du WMS comme fenêtre de prévisualisation OpenLayers. <a href="#">OpenLayers</a> est une bibliothèque JavaScript open source pour afficher des cartes dans des navigateurs web. La sortie OpenLayers possède des filtres avancés qui ne sont pas disponibles lors de l'utilisation seule d'OpenLayers. De plus, la prévisualisation générée contient un en-tête qui facilite les options de configuration pour l'affichage.
PDF	Un fichier PDF (Portable Document Format) encapsule une description complète d'une mise en page définie d'un document 2D, dont du texte, des polices des images raster et des graphiques vecteurs 2D.



Exemple de sortie image - une prévisualisation OpenLayers de la couche nurc:Pk50095

#### **Sortie texte**

Format	Description
AtomPub	Sortie WMS de données spatiales au format XML. Le format AtomPub (Atom Publishing Protocol) est un protocole d'application pour publier et éditer des ressources web en utilisant le protocol HTTP et le langage XML. Développé pour remplacer la famille des standarads RSS pour la syndication de contenu, Atom permet la subscription de données géographiques.
GeoRss	Sortie de la requête GetMap du WMS de données vecteurs au format XML. Le format RSS (Rich Site Summary) est un format XML pour délivrer régulièrement du contenu web modifiée. GeoRss est un standard pour encoder la localisation sous forme de flux RSS. La prévisualisation de couches produit des documents RSS 2.0 avec des géométries simples GeoRSS en utilisant Atom.

## Une réponse efficace à la directive INSPIRE

GeoJSON	JavaScript Object Notation (JSON) est un format léger d'échange de données basé sur le langage de programmation JavaScript. Cela en fait un format d'échange idéal pour les applications basées sur les navigateurs puisqu'il peut être lu directement et facilement en JavaScript. GeoJSON est un format de sortie en texte qui ajoute des types géographiques au JSON.
CSV	Sortie du GetFeature du WFS en format textuel séparé par des virgules. Les fichiers CSV (Comma Separated Values) sont des fichiers textes contenant des lignes de données. Les valeurs de données dans chaque ligne sont séparées par des virgules. Les fichiers CSV contiennent aussi un en-tête séparé par des virgules dénommant chaque valeur des lignes. Les CSV de GeoServer sont entièrement en streaming, avec aucune limitation de la quantité de données qui peuvent être renvoyées.

Un échantillon d'un GeoRSS simple pour la couche nurc:Pk50095 en utilisant Atom :

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:atom="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss" version="2.0">
  <channel>
    <title>Pk50095</title>
    <description>Feed auto-generated by GeoServer</description>
    <link></link>
    <item>
      <title>fid--f04ca6b_1226f8d829e_-7ff4</title>
      <georss:polygon>46.722110379286 13.00635746384126
        46.72697223230676 13.308182612644663 46.91359611878293
        13.302316867622581 46.90870264238999 12.999446822650462
        46.722110379286 13.00635746384126
      </georss:polygon>
    </item>
  </channel>
</rss>
```

## Sortie de données

Toutes les sorties de données sont réalisées par une requête GetFeature du WFS sur des données vecteurs.

Format	Description
GML2/3	GML (Geography Markup Language) is the XML grammar defined by the <a href="#">Open Geospatial Consortium</a> (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic data sharing. GML2 is the default (Common) output format, while GML3 is available from the "All Formats" drop down menu.
Shapefile	The ESRI Shapefile or simply a shapefile is the most commonly used format for exchanging GIS data. GeoServer outputs shapefiles in zip format, with a directory of .cst, .dbf, .prg, .shp, and .shx files.

## Symbolisation

Cette section traite de la symbolisation des données spatiales servies par GeoServer.

## Introduction à SLD

Les données géospatiales n'ont pas de composante visuelle intrinsèque. Pour voir une donnée, elle doit être symbolisée. Cela signifie préciser couleur, épaisseur et autres attributs visibles. Dans GeoServer, cette symbolisation est effectuée en utilisant un langage balisé appelé [Styled Layer Descriptor](#), ou SLD pour faire court. SLD est un langage balisé basé sur XML et est très puissant, mais il peut être intimidant. Cette page fournit des instructions élémentaires sur ce que l'on peut faire avec SLD et comment GeoServer le prend en charge.

### Note

Comme GeoServer utilise exclusivement SLD pour la symbolisation (ou "stylage", NdT), les termes "SLD" et "styles" seront utilisés indifféremment.

## **Types de symbolisation**

Les données que GeoServer peut servir sont réparties en trois classes de forme: **Points, lignes et polygones**. Les lignes (des formes unidimensionnelles) sont les plus simples, car elles n'ont que des arêtes (également appelé "stroke" pour "trait") à symboliser. Les polygones, des formes bidimensionnelles, ont des arêtes et un intérieur (également appelé "fill" pour "remplissage"). Les points, même s'il n'ont pas de dimension, ont également des arêtes et un intérieur (et une taille) qui peuvent être symbolisés. Pour les remplissages, on peut spécifier une couleur; pour les traits, couleur et épaisseur peuvent être spécifiés.

Il est possible de fabriquer des symbolisations plus élaborées que couleur et trait. Les points peuvent être représentés avec des formes connues telles que cercles, carrés, étoiles, et même des graphismes personnalisés ou du texte. Les lignes peuvent être symbolisées avec des pointillés ou des hachures. Les polygones peuvent être remplis avec des motifs graphiques personnalisés. Les styles peuvent être basés sur des attributs dans les données, pour que certains objets soient représentés différemment. Les labels textuels sur les objets sont également possibles. Les objets peuvent être symbolisés différemment selon le niveau de zoom, avec la taille de l'objet déterminant la façon dont il est représenté. Les possibilités sont vastes.

## **Métadonnées de style**

### **GeoServer et SLD**

Toute couche (featuretype) enregistrée dans GeoServer a besoin d'avoir au moins un style associé. GeoServer est livré avec quelques styles simples, et il est possible d'y adjoindre autant de styles que souhaité. Il est possible de modifier à tout moment le style associé à une couche dans la page [Couches](#) de [Interface Web d'administration](#). Lorsqu'on souhaite ajouter simultanément une couche et un style à GeoServer, le style devrait être ajouté en premier, pour que la nouvelle couche puisse être associée immédiatement au style. Vous pouvez ajouter un style dans le menu [Styles](#) de [Interface Web d'administration](#).

### **Définitions**

#### **Symbolizer**

#### **Rule**

#### **FeatureTypeStyle**

#### **Un style simple**

## Une réponse efficace à la directive INSPIRE

Ce SLD prend une couche contenant des points, et les symbolise sous forme de cercles rouges avec une taille de 6 pixels. (C'est le premier exemple dans la section *Points* de *Livre de recettes SLD*.)

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <StyledLayerDescriptor version="1.0.0"
3   xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
4   xmlns="http://www.opengis.net/sld"
5   xmlns:ogc="http://www.opengis.net/ogc"
6   xmlns:xlink="http://www.w3.org/1999/xlink"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8   <NamedLayer>
9     <Name>Simple point</Name>
10    <UserStyle>
11      <Title>GeoServer SLD Cook Book: Simple point</Title>
12      <FeatureTypeStyle>
13        <Rule>
14          <PointSymbolizer>
15            <Graphic>
16              <Mark>
17                <WellKnownName>circle</WellKnownName>
18                <Fill>
19                  <CssParameter name="fill">#FF0000</CssParameter>
20                </Fill>
21              </Mark>
22              <Size>6</Size>
23            </Graphic>
24          </PointSymbolizer>
25        </Rule>
26      </FeatureTypeStyle>
27    </UserStyle>
28  </NamedLayer>
29 </StyledLayerDescriptor>
```

La longueur de cet exemple simple ne doit pas vous intimider; seules quelques lignes sont réellement utiles à la compréhension. La **ligne 14** dit que nous utilisons un symboliseur ponctuel ("PointSymbolizer"), un style pour les données ponctuelles. La **ligne 17** signifie que nous allons utiliser un "well known name" (un nom connu), un cercle, pour représenter les points. Il y a beaucoup de noms connus pour les formes, comme "square" (carré), "star" (étoile), "triangle", etc. Les **lignes 18-20** signifient que l'on remplit la forme avec une couleur valant #FF0000 (rouge). C'est un code couleur RGB, écrit en hexadécimal, sous la forme #RRVVBB. Finalement, la **ligne 22** signifie que la taille de la forme est de 6 pixels en largeur. Le reste de la structure contient des métadonnées à propos du style, comme Name/Title/Abstract (nom/titre/résumé).

On trouve bien d'autres exemples dans le *Livre de recettes SLD*.

### Note

Vous vous apercevrez que certaines balises portent un préfixe, comme ogc:. Ces préfixes matérialisent des **espaces de nom XML**. Dans les balises des **lignes 2-7**, il y a deux espaces de nom XML, l'un appelé xmlns, et l'autre appelé xmlns:ogc. Les balises correspondant au premier espace de nom ne requièrent pas de préfixe,

mais celles correspondant au deuxième requièrent le préfixe `ogc:`. Il faut mentionner que le nom des espaces de nom n'est pas important: le premier espace de nom pourrait être `xmlns:sld` (comme c'est l'usage) et ainsi toutes les balises de cet exemple devraient avoir pour préfixe `sld:`. L'important est qu'à l'espace de nom corresponde bien les balises associées.

## Dépannage

SLD est une forme de langage de programmation, peut différent de la création d'une page web ou de la rédaction d'un script. En conséquence, vous pouvez rencontrer des problèmes qui auront besoin d'un dépannage. Lorsqu'un style est ajouté à GeoServer, il est automatiquement soumis à validation selon la spécification OGC SLD (cette vérification peut être court-circuite), mais les erreurs ne seront pas traquées. Il est très facile d'avoir des erreurs de syntaxe dissimulées dans un SLD valide. La plupart du temps, cela occasionnera une carte ne présentant aucun objet (une carte blanche), mais quelquefois des erreurs iront jusqu'à bloquer le chargement de la carte.

La façon la plus facile de corriger les erreurs dans un SLD est de les isoler. Si le SLD est long et comprend de nombreux filtres et règles différents, essayez de supprimer temporairement certains d'entre eux pour voir si les erreurs disparaissent.

Pour réduire les erreurs lors de la création d'un SLD, il est recommandé d'utiliser un éditeur de texte prévu pour travailler avec XML. Les éditeurs adaptés à XML savent rendre la recherche et la suppression d'erreur plus facile en fournissant de la coloration syntaxique et (parfois) une détection d'erreur intégrée.

## Livre de recettes SLD

Le livre de recettes SLD est une collection de "recettes SLD" pour créer une variété de styles cartographiques. Autant que possible, chaque exemple est conçu pour montrer une seule fonctionnalité de SLD pour que le code puisse être copié de l'exemple et adapté lorsque vous créez vos propres SLD. Quoique ne représentant pas une référence exhaustive comme le [SLD Reference](#) ou la [spécification OGC SLD 1.0](#) le livre recettes SLD est conçu pour devenir une référence pratique, recensant des modèles de styles simples à comprendre.

Le livre de recettes SLD est divisé en quatre sections : les trois premières pour chacun des types vecteur (points, lignes, polygones) et la quatrième pour les rasters. Dans toute section, chaque exemple présente une capture d'écran montrant la sortie WMS de GeoServer, un extrait du code SLD pour référence, et un lien pour télécharger le SLD complet.

Chaque section utilise des données fabriquées spécialement pour le livre de recettes SLD, avec des shapefiles pour les données vecteur et des GeoTIFFs pour les données raster. La projection des données est EPSG:4326. Tous les fichiers peuvent être aisément chargés dans GeoServer pour faire fonctionner les exemples.

Data Type	Shapefile
Point	<a href="#">sld_cookbook_point.zip</a>
Ligne	<a href="#">sld_cookbook_line.zip</a>
Polygone	<a href="#">sld_cookbook_polygon.zip</a>
Raster	<a href="#">sld_cookbook_raster.zip</a>

## Points

## Une réponse efficace à la directive INSPIRE

Si les points paraissent être le type de forme le plus simple, ne comprenant qu'une position et pas d'autres dimensions, il existe de nombreuses façons différentes de symboliser un point en SLD.

### Avis

Pour rester concis, les exemples de code présentés sur cette page ne sont **pas le code SLD complet** car ils omencent les informations SLD de début et de fin. Utilisez les liens pour télécharger les SLD complets de chaque exemple.

### Exemple de couche ponctuelle

La couche ponctuelle utilisée pour les exemples ci-dessous contient nom et population des villes principales d'un pays fictif. Pour mémoire, la table d'attributs des points de cette couche est présentée ci-dessous.

<b>fid</b> (Feature ID)	<b>name</b> (City name)	<b>pop</b> (Population)
point.1	Borfin	157860
point.2	Supox City	578231
point.3	Ruckis	98159
point.4	Thisland	34879
point.5	Synopolis	24567
point.6	San Glissando	76024
point.7	Detrania	205609

Téléchargez le shapefile des points

#### **Point simple**

Cet exemple symbolise les points sous la forme de cercles rouges de diamètre 6 pixels.



*Simple point*

### Code

Voir et télécharger le SLD "Simple point" complet

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PointSymbolizer>
4        <Graphic>
5          <Mark>
6            <WellKnownName>circle</WellKnownName>
7            <Fill>
8              <CssParameter name="fill">#FF0000</CssParameter>
9            </Fill>
10           </Mark>
11           <Size>6</Size>
12         </Graphic>
13       </PointSymbolizer>
14     </Rule>
15   </FeatureTypeStyle>
```

### Details

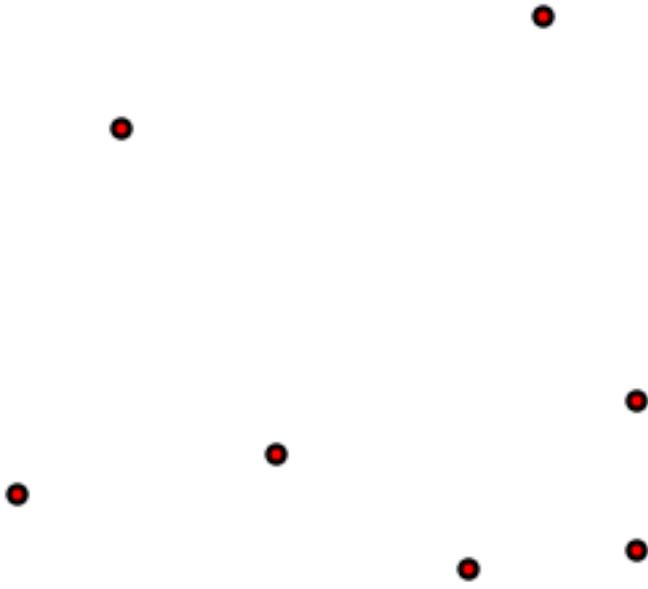
Il y a un `<Rule>` dans un `<FeatureTypeStyle>` pour ce SLD, ce qui est la situation la plus simple possible. (Les exemples suivants contiendront un `<Rule>` et un `<FeatureTypeStyle>` sauf précision.) La symbolisation des points est effectuée par le `<PointSymbolizer>` (**lignes 3-13**). La **ligne 6** dit que la forme du symbole doit être un cercle, avec la **ligne 8** fixant la couleur de remplissage à rouge (#FF0000). La **ligne 11** fixe la taille (diamètre) du graphisme

Une réponse efficace à la directive INSPIRE

à 6 pixels.

### **Exemple simple avec bord**

Cet exemple ajoute un trait (ou bord) autour du :*sld\_cookbook\_points\_simplepoint*, avec le trait coloré en noir et muni d'une épaisseur de 2 pixels.



*Point simple avec bord*

### **Code**

Consultez et téléchargez le SLD complet "Point simple avec bord"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PointSymbolizer>
4        <Graphic>
5          <Mark>
6            <WellKnownName>circle</WellKnownName>
7            <Fill>
8              <CssParameter name="fill">#FF0000</CssParameter>
9            </Fill>
10           <Stroke>
11             <CssParameter name="stroke">#000000</CssParameter>
12             <CssParameter name="stroke-width">2</CssParameter>
13           </Stroke>
14         </Mark>
15         <Size>6</Size>
16       </Graphic>
17     </PointSymbolizer>
```

Une réponse efficace à la directive INSPIRE

```
18      </Rule>
19  </FeatureTypeStyle>
```

### Détails

Cet exemple est similaire à l'exemple [Point simple](#). Les **lines 10-13** spécifient le trait, avec la **ligne 11** réglant couleur à noir (#000000) et la **ligne 12** réglant l'épaisseur à 2 pixels.

### Carré avec rotation

Cet exemple crée un carré au lieu d'un cercle, le colore en vert, le dimensionne à 12 pixels et le fait tourner de 45 degrés.



*Carré avec rotation*

### Code

Consultez et téléchargez le SLD complet "Carré avec rotation"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PointSymbolizer>
4        <Graphic>
5          <Mark>
6            <WellKnownName>square</WellKnownName>
7            <Fill>
8              <CssParameter name="fill">#009900</CssParameter>
9              </Fill>
10         </Mark>
```

Une réponse efficace à la directive INSPIRE

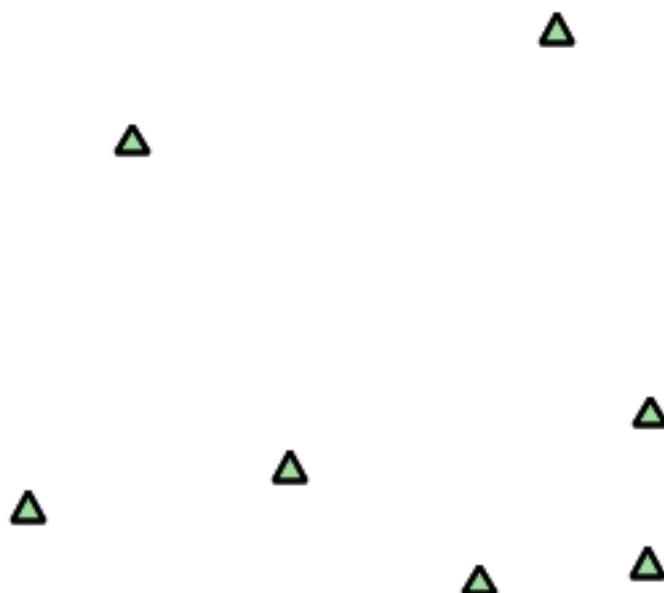
```
11      <Size>12</Size>
12      <Rotation>45</Rotation>
13    </Graphic>
14  </PointSymbolizer>
15 </Rule>
16 </FeatureTypeStyle>
```

### Détails

Dans cet exemple, la **ligne 6** demande pour forme un carré, avec la **ligne 8** réglant la couleur à vert foncé (#009900). La **ligne 11** règle la taille à 12 pixels et la **ligne 12** règle la rotation à 45 degrés.

### Triangle transparent

Cet exemple dessine un triangle, crée un trait noir identique à l'exemple [Exemple simple avec bord](#), et règle le remplissage du triangle à 20% d'opacité (presque transparent).



*Triangle transparent*

### Code

Consultez et téléchargez le SLD complet "Triangle transparent" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PointSymbolizer>
4        <Graphic>
5          <Mark>
```

```
6      <WellKnownName>triangle</WellKnownName>
7      <Fill>
8          <CssParameter name="fill">#009900</CssParameter>
9          <CssParameter name="fill-opacity">0.2</CssParameter>
10     </Fill>
11     <Stroke>
12         <CssParameter name="stroke">#000000</CssParameter>
13         <CssParameter name="stroke-width">2</CssParameter>
14     </Stroke>
15     </Mark>
16     <Size>12</Size>
17     </Graphic>
18   </PointSymbolizer>
19 </Rule>
20 </FeatureTypeStyle>
```

### Détails

Dans cet exemple, la **ligne 6** règle la forme, un triangle dans ce cas. La **ligne 8** règle la couleur de remplissage à vert foncé (#009900) et la **ligne 9** règle l'opacité à 0.2 (opaque à 20%). Une valeur d'opacité de 1 signifie que la forme est dessinée avec une opacité de 100%, alors qu'une valeur d'opacité de 0 signifie que la forme est dessinée avec une opacité de 0%, soit complètement transparente. La valeur de 0.2 (20% d'opacité) signifie que le remplissage des points prend partiellement la couleur et le style de ce qui est dessiné en-dessous. Dans cet exemple, comme le fond est blanc, le vert foncé paraît plus clair. Si les points étaient posés sur fond noir, la couleur résultante serait plus foncée. Les **lignes 12-13** règlent la couleur de trait à noir (#000000) et la largeur à 2 pixels. Finalement, la **ligne 16** règle la taille du point à un diamètre de 12 pixels.

### Point graphique

Cet exemple symbolise chaque point avec un graphisme au lieu d'une forme simple.



*Point image*

### **Code**

Consultez et téléchargez le SLD complet "Point image"

```
1   <FeatureTypeStyle>
2     <Rule>
3       <PointSymbolizer>
4         <Graphic>
5           <ExternalGraphic>
6             <OnlineResource
7               xlink:type="simple"
8               xlink:href="smileyface.png" />
9             <Format>image/png</Format>
10            </ExternalGraphic>
11            <Size>32</Size>
12          </Graphic>
13        </PointSymbolizer>
14      </Rule>
15    </FeatureTypeStyle>
```

### **Détails**

Ce style utilise une image au lieu d'une forme simple pour représenter les points. Dans le SLD, ceci est connu comme un `<ExternalGraphic>`, pour le distinguer des formes courantes comme carrés et cercles qui sont "internes" au moteur de rendu. Les **lignes 5-10** spécifient les détails de cette image. La **ligne 8** paramètre le chemin et le nom de fichier de l'image, alors que la **ligne 9** indique le format (type MIME) de l'image (`image/png`). Dans cet

## Une réponse efficace à la directive INSPIRE

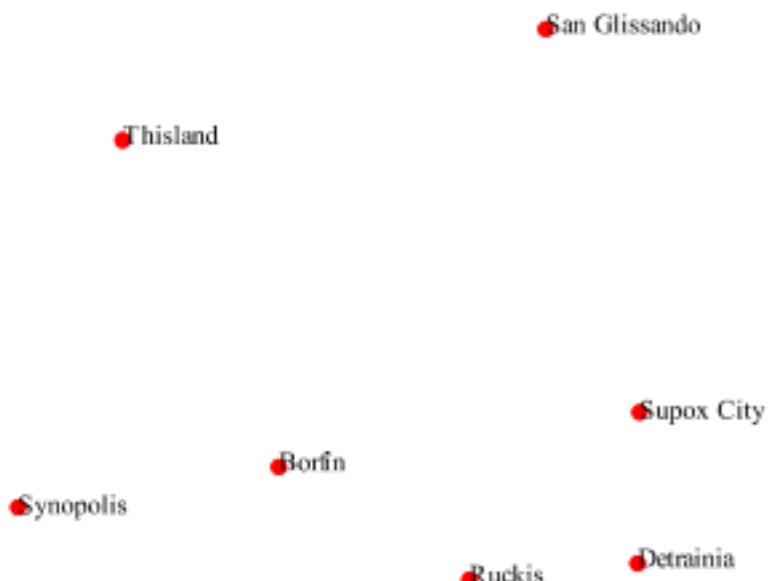
exemple, l'image est contenue dans le même répertoire que le SLD, aucune information de chemin n'est donc nécessaire en **ligne 8**, mais une URL complète pourrait être utilisée si souhaité. La **ligne 11** détermine la taille d'affichage de l'image; ceci peut être réglé indépendamment de la dimension propre de l'image, même si dans ce cas elles sont identiques (32 pixels). Si l'image était rectangulaire, la valeur <Size> s'appliquerait à la *hauteur* seule de l'image, avec la largeur réévaluée proportionnellement.



*Graphisme utilisé pour un point*

### **Point avec étiquette par défaut**

Cet exemple présente une étiquette textuelle sur le *Point simple* affichant l'attribut "name" du point. Une étiquette sera représentée de cette façon en l'absence de personnalisation.



*Point avec étiquette par défaut*

### **Code**

Consultez et téléchargez le SLD complet "Point with default label"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PointSymbolizer>
4        <Graphic>
5          <Mark>
6            <WellKnownName>circle</WellKnownName>
7            <Fill>
```

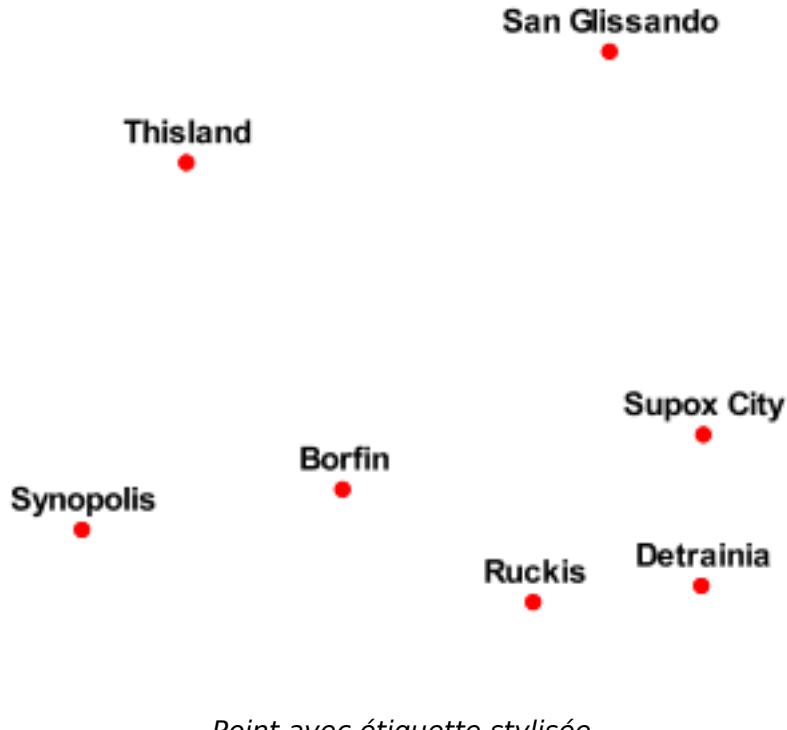
```
8          <CssParameter name="fill">#FF0000</CssParameter>
9          </Fill>
10         <Mark>
11             <Size>6</Size>
12         </Graphic>
13     </PointSymbolizer>
14     <TextSymbolizer>
15         <Label>
16             <ogc:PropertyName>name</ogc:PropertyName>
17         </Label>
18         <Fill>
19             <CssParameter name="fill">#000000</CssParameter>
20         </Fill>
21     </TextSymbolizer>
22   </Rule>
23 </FeatureTypeStyle>
```

### Détails

Les **lignes 3-13**, qui contiennent le `<PointSymbolizer>`, sont identiques à l'exemple *Point simple* ci-dessus. L'étiquette est paramétrée dans le `<TextSymbolizer>` aux **lignes 14-27**. Les **Lignes 15-17** déterminent le texte à afficher dans l'étiquette, dans ce cas la valeur de l'attribut "name". (Consultez la table des attributs dans la section *Exemple de couche ponctuelle* si nécessaire.) La **ligne 19** règle la couleur. Tous les autres paramètres concernant l'étiquette sont réglés aux valeurs par défaut du moteur de rendu, c'est à dire police Times New Roman, couleur noire, et taille de police 10 pixels. Le coin bas gauche de l'étiquette est aligné avec le centre du point.

### **Point avec étiquette stylisée**

Cet exemple améliore le style de l'étiquette de l'exemple *Point avec étiquette par défaut* en centrant l'étiquette au-dessus du point et en indiquant un nom de police et une taille différents.



*Point avec étiquette stylisée*

### Code

Consultez et téléchargez le SLD complet "Point avec étiquette stylisée"

```
1   <FeatureTypeStyle>
2     <Rule>
3       <PointSymbolizer>
4         <Graphic>
5           <Mark>
6             <WellKnownName>circle</WellKnownName>
7             <Fill>
8               <CssParameter name="fill">#FF0000</CssParameter>
9             </Fill>
10            <Mark>
11              <Size>6</Size>
12            </Mark>
13          </Graphic>
14        <PointSymbolizer>
15          <Label>
16            <ogc:PropertyName>name</ogc:PropertyName>
17          </Label>
18          <Font>
19            <CssParameter name="font-family">Arial</CssParameter>
20            <CssParameter name="font-size">12</CssParameter>
21            <CssParameter name="font-style">normal</CssParameter>
22            <CssParameter name="font-weight">bold</CssParameter>
23          </Font>
```

```
24      <LabelPlacement>
25          <PointPlacement>
26              <AnchorPoint>
27                  <AnchorPointX>0.5</AnchorPointX>
28                  <AnchorPointY>0.0</AnchorPointY>
29              </AnchorPoint>
30          <Displacement>
31              <DisplacementX>0</DisplacementX>
32              <DisplacementY>5</DisplacementY>
33          </Displacement>
34      </PointPlacement>
35  </LabelPlacement>
36  <Fill>
37      <CssParameter name="fill">#000000</CssParameter>
38  </Fill>
39  </TextSymbolizer>
40 </Rule>
41 </FeatureTypeStyle>
```

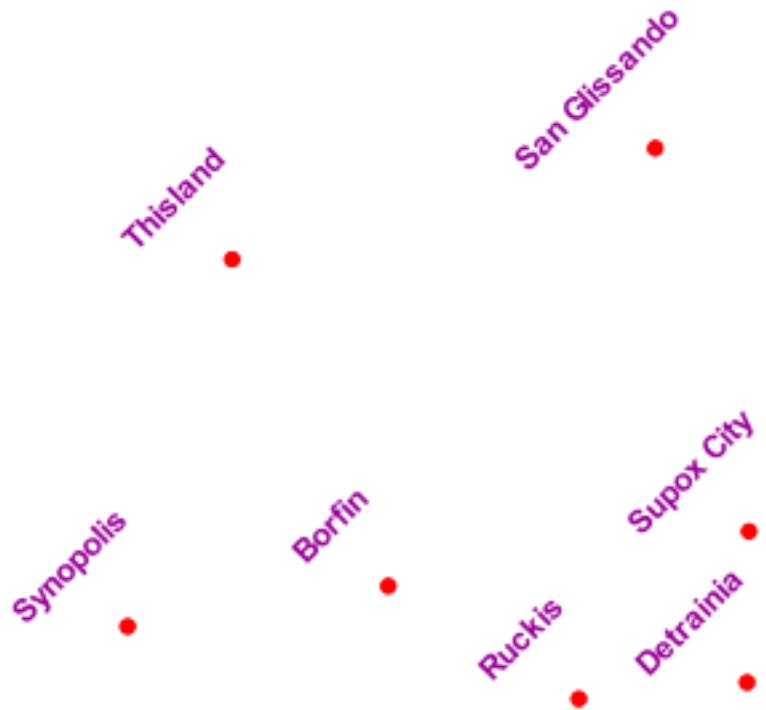
### Détails

Dans cet exemple, les **lignes 3-13** sont identiques à l'exemple *Point simple* ci-dessus. Le *<TextSymbolizer>* au lignes 14-39 contient beaucoup d'autres détails concernant le style de l'étiquette que dans l'exemple précédent *Point avec étiquette par défaut*. Les **lignes 15-17** indique à nouveau l'emploi de l'attribut "name" pour afficher du texte. Les **lignes 18-23** règlent la police: la **ligne 19** paramètrent la famille de police à "Arial", la **ligne 20** paramètre la taille de font à 12, la **ligne 21** paramètre le style de police à "normal" (par opposition à "italic" ou "oblique"), et la **ligne 22** règle le poids de police à gras ou "bold" (par opposition à "normal"). Les **lignes 24-35** (*<LabelPlacement>*) déterminent le placement de l'étiquette par rapport au point. *<AnchorPoint>* (**lignes 26-29**) paramètre le point d'intersection entre l'étiquette et le point, avec ici (**ligne 27-28**) le point centré (0.5) horizontalement et verticalement aligné avec le bas (0.0) de l'étiquette. Il y a aussi *<Displacement>* (**lignes 30-33**), qui règle le décalage de l'étiquette relativement à la ligne, dans ce cas 0 pixels horizontalement (**ligne 31**) et 5 pixels verticalement (**ligne 32**). Finalement, la **ligne 37** règle la couleur de police de l'étiquette à noir (#000000).

Le résultat est une étiquette centrée et en gras placée légèrement au-dessus de chaque point.

### **Point avec rotation de l'étiquette**

Cet exemple est construit sur la base de l'exemple précédent *Point avec étiquette stylisée* en faisant tourner l'étiquette de 45 degrés, en positionnant les étiquettes plus loin des points, et en modifiant la couleur de l'étiquette à pourpre.



*Point avec la rotation de l'étiquette*

### Code

Consultez et téléchargez le SLD complet "Point avec la rotation de l'étiquette"

```
1   <FeatureTypeStyle>
2     <Rule>
3       <PointSymbolizer>
4         <Graphic>
5           <Mark>
6             <WellKnownName>circle</WellKnownName>
7             <Fill>
8               <CssParameter name="fill">#FF0000</CssParameter>
9             </Fill>
10            </Mark>
11            <Size>6</Size>
12          </Graphic>
13        </PointSymbolizer>
14        <TextSymbolizer>
15          <Label>
16            <ogc:PropertyName>name</ogc:PropertyName>
17          </Label>
18          <Font>
19            <CssParameter name="font-family">Arial</CssParameter>
20            <CssParameter name="font-size">12</CssParameter>
21            <CssParameter name="font-style">normal</CssParameter>
22            <CssParameter name="font-weight">bold</CssParameter>
23          </Font>
```

```
24      <LabelPlacement>
25          <PointPlacement>
26              <AnchorPoint>
27                  <AnchorPointX>0.5</AnchorPointX>
28                  <AnchorPointY>0.0</AnchorPointY>
29              </AnchorPoint>
30          <Displacement>
31              <DisplacementX>0</DisplacementX>
32              <DisplacementY>25</DisplacementY>
33          </Displacement>
34          <Rotation>-45</Rotation>
35      </PointPlacement>
36  </LabelPlacement>
37  <Fill>
38      <CssParameter name="fill">#990099</CssParameter>
39  </Fill>
40      <TextSymbolizer>
41          <Rule>
42      </TextSymbolizer>
43  </FeatureTypeStyle>
```

### Détails

Cet exemple est similaire à [Point avec étiquette stylisée](#), mais il y a trois différences importantes. La **ligne 32** spécifie un déplacement vertical de 25 pixels. La **ligne 34** spécifie une rotation de "-45", ou 45 degrés dans le sens inverse des aiguilles d'une montre. (Les valeurs de rotation vont en augmentant dans le sens des aiguilles d'une montre, c'est pourquoi cette valeur est négative.) Finalement, la **ligne 38** paramètre la couleur de police à une nuance de pourpre (#990099).

Notez que, pendant le rendu, le déplacement prend effet avant la rotation, donc dans cet exemple le déplacement de 25 pixels en vertical subit lui-même une rotation de 45 degrés.

### Points basés sur les attributs

Cet exemple fait varier la taille du symbole selon la valeur de l'attribut population ("pop").



*Points basés sur les attributs*

### **Code**

Consultez et téléchargez le SLD complet "Points basés sur les attributs"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <Name>SmallPop</Name>
4      <Title>1 to 50000</Title>
5      <ogc:Filter>
6        <ogc:PropertyIsLessThan>
7          <ogc:PropertyName>pop</ogc:PropertyName>
8          <ogc:Literal>50000</ogc:Literal>
9        </ogc:PropertyIsLessThan>
10       </ogc:Filter>
11       <PointSymbolizer>
12         <Graphic>
13           <Mark>
14             <WellKnownName>circle</WellKnownName>
15             <Fill>
16               <CssParameter name="fill">#0033CC</CssParameter>
17             </Fill>
18           </Mark>
19           <Size>8</Size>
20         </Graphic>
21       </PointSymbolizer>
22     </Rule>
23   <Rule>
```

## Une réponse efficace à la directive INSPIRE

```
24      <Name>MediumPop</Name>
25      <Title>50000 to 100000</Title>
26      <ogc:Filter>
27          <ogc:And>
28              <ogc:PropertyIsGreaterThanOrEqualTo>
29                  <ogc:PropertyName>pop</ogc:PropertyName>
30                  <ogc:Literal>50000</ogc:Literal>
31              </ogc:PropertyIsGreaterThanOrEqualTo>
32              <ogc:PropertyIsLessThan>
33                  <ogc:PropertyName>pop</ogc:PropertyName>
34                  <ogc:Literal>100000</ogc:Literal>
35              </ogc:PropertyIsLessThan>
36          </ogc:And>
37      </ogc:Filter>
38      <PointSymbolizer>
39          <Graphic>
40              <Mark>
41                  <WellKnownName>circle</WellKnownName>
42                  <Fill>
43                      <CssParameter name="fill">#0033CC</CssParameter>
44                  </Fill>
45              </Mark>
46              <Size>12</Size>
47          </Graphic>
48      </PointSymbolizer>
49  </Rule>
50  <Rule>
51      <Name>LargePop</Name>
52      <Title>Greater than 100000</Title>
53      <ogc:Filter>
54          <ogc:PropertyIsGreaterThanOrEqualTo>
55              <ogc:PropertyName>pop</ogc:PropertyName>
56              <ogc:Literal>100000</ogc:Literal>
57          </ogc:PropertyIsGreaterThanOrEqualTo>
58      </ogc:Filter>
59      <PointSymbolizer>
60          <Graphic>
61              <Mark>
62                  <WellKnownName>circle</WellKnownName>
63                  <Fill>
64                      <CssParameter name="fill">#0033CC</CssParameter>
65                  </Fill>
66              </Mark>
67              <Size>16</Size>
68          </Graphic>
69      </PointSymbolizer>
70  </Rule>
71  </FeatureTypeStyle>
```

### Détails

### Note

Se référer à [Exemple de couche ponctuelle](#) pour voir les attributs des données. Cet exemple a mis de côté les étiquettes pour simplifier le style, mais vous pouvez vous référer à l'exemple [Point avec étiquette stylisée](#) pour voir à quels attributs correspondent quels points.

Le style contient trois règles. Chaque <Rule> fait varier le style de chaque point selon la valeur de l'attribut population ("pop"), de plus petites valeurs ayant pour effet des cercles plus petits, et les valeurs plus grandes un cercle plus grand.

Les trois règles sont conçues comme suit:

Rule order	Rule name	Population ("pop")	Size
1	SmallPop	Less than 50,000	8
2	MediumPop	50,000 to 100,000	12
3	LargePop	Greater than 100,000	16

L'ordre des règles n'a pas d'importance dans ce cas, car chaque forme est représentée par une seule règle.

La première règle, en **lignes 2-22**, spécifie la symbolisation des points dont l'attribut population est inférieur à 50 000. Les **lignes 5-10** règlent ce filtre, avec les **lignes 6-9** déterminant le filtre "inférieur à", la **ligne 7** précisant l'attribut ("pop"), et la **ligne 8** une valeur de 50 000. Le symbole est un cercle (**ligne 14**), la couleur est bleu foncé (#0033CC, en **ligne 16**), et la taille est 8 pixels en diamètre (**ligne 19**).

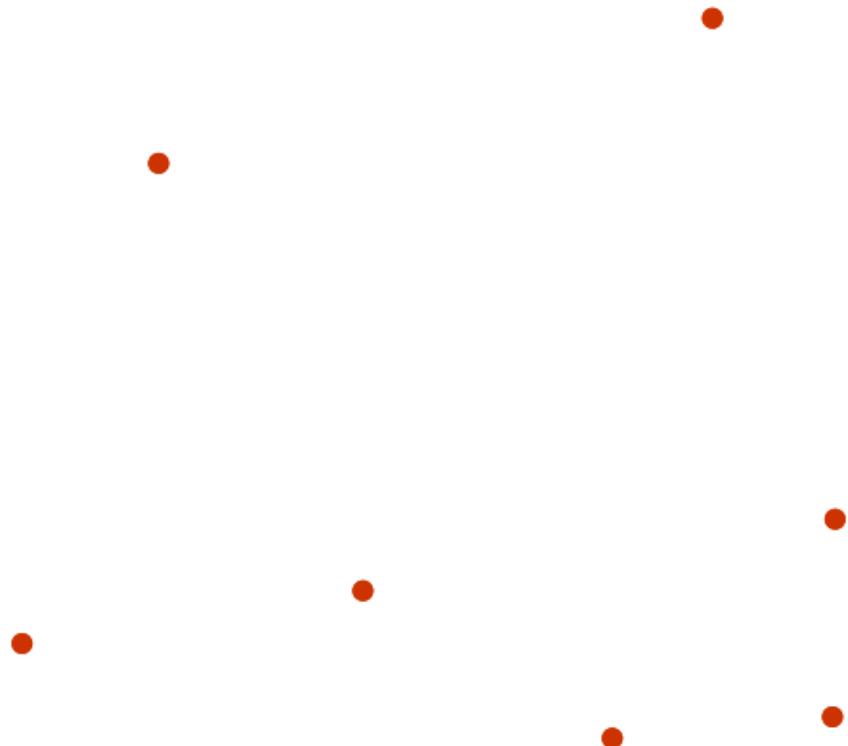
La deuxième règle, en **lignes 23-49**, spécifie une symbolisation pour les points dont l'attribut population est supérieur ou égal à 50 000 et inférieur à 100 000. Le filtre sur la population est réglé en **lignes 26-37**. Ce filtre est plus long que la première règle car deux critères au lieu d'un doivent être employés: un filtre "supérieur ou égal à" et un filtre "inférieur à". Notez le And en **ligne 27** et **ligne 36**. Ceci induit que les deux filtres doivent être vrais pour que la règle s'applique. La taille de la forme est réglée à 12 pixels en **ligne 46**. Toutes les autres directives de style sont identiques à la première règle.

La troisième règle, en **lignes 50-70**, spécifie une symbolisation pour les points dont l'attribut population est supérieur ou égal à 100 000. Le filtre sur la population est réglé en **lignes 53-58**, et la seule autre différence est la taille du cercle, à 16 pixels (**ligne 67**) dans cette règle.

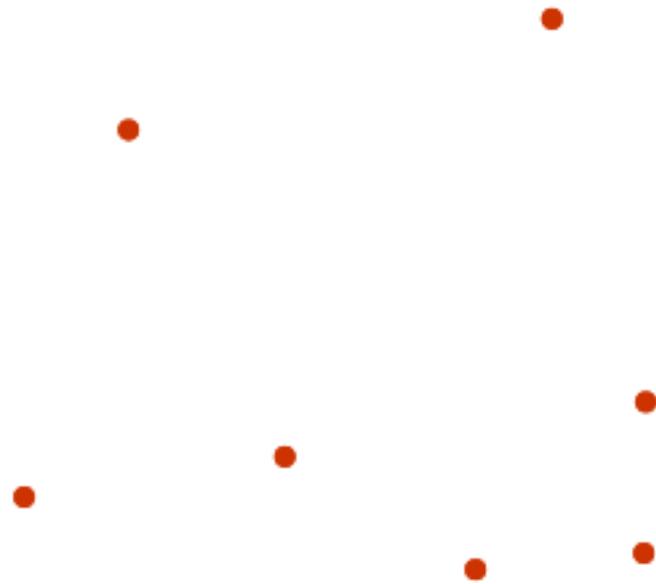
ce style a pour résultat que les cités avec une population plus grande ont des points plus grands.

### Points basés sur le zoom

Cet exemple modifie la symbolisation des points à des niveaux de zoom différents



*Zoom-based point: Zoomed in*



*Zoom-based point: Partially zoomed*



*Zoom-based point: Zoomed out*

Une réponse efficace à la directive INSPIRE

### Code

Téléchargez et consultez le SLD "Points basés sur le zoom"

```
1  <FeatureTypeStyle>
2      <Rule>
3          <Name>Large</Name>
4          <MaxScaleDenominator>160000000</MaxScaleDenominator>
5          <PointSymbolizer>
6              <Graphic>
7                  <Mark>
8                      <WellKnownName>circle</WellKnownName>
9                      <Fill>
10                         <CssParameter name="fill">#CC3300</CssParameter>
11                     </Fill>
12                 </Mark>
13                 <Size>12</Size>
14             </Graphic>
15         </PointSymbolizer>
16     </Rule>
17     <Rule>
18         <Name>Medium</Name>
19         <MinScaleDenominator>160000000</MinScaleDenominator>
20         <MaxScaleDenominator>320000000</MaxScaleDenominator>
21         <PointSymbolizer>
22             <Graphic>
23                 <Mark>
24                     <WellKnownName>circle</WellKnownName>
25                     <Fill>
26                         <CssParameter name="fill">#CC3300</CssParameter>
27                     </Fill>
28                 </Mark>
29                 <Size>8</Size>
30             </Graphic>
31         </PointSymbolizer>
32     </Rule>
33     <Rule>
34         <Name>Small</Name>
35         <MinScaleDenominator>320000000</MinScaleDenominator>
36         <PointSymbolizer>
37             <Graphic>
38                 <Mark>
39                     <WellKnownName>circle</WellKnownName>
40                     <Fill>
41                         <CssParameter name="fill">#CC3300</CssParameter>
42                     </Fill>
43                 </Mark>
44                 <Size>4</Size>
45             </Graphic>
46         </PointSymbolizer>
47     </Rule>
48 </FeatureTypeStyle>
```

### Détails

## Une réponse efficace à la directive INSPIRE

On souhaite souvent rendre les formes plus grandes avec les hauts niveaux de zooms pour créer une carte à l'apparence naturelle. Cet exemple symbolise les points de façon à faire varier leur taille selon le niveau de zoom (ou plus exactement, le dénominateur d'échelle). Les dénominateurs d'échelle correspondent à l'échelle de la carte. Un dénominateur d'échelle de 10 000 signifie que la carte a une échelle de 1/10 000e en unités associées à la projection de la carte.

### Note

Déterminer des dénominateurs d'échelle (niveaux de zoom) pertinents va au-delà de cet exemple.

Le style contient trois règles. Les trois règles sont conçues comme suit:

Rule order	Rule name	Scale denominator	Point size
1	Large	1:160,000,000 or less	12
2	Medium	1:160,000,000 to 1:320,000,000	8
3	Small	Greater than 1:320,000,000	4

L'ordre de ces règles n'a pas d'importance car les échelles définies dans chaque règle ne se recoupent pas.

La première règle (**lignes 2-16**) correspond au plus petit dénominateur d'échelle, lorsque la vue est "zoomée". Le seuil d'échelle est réglé en **ligne 4**, pour que la règle s'applique à toute carte avec un dénominateur d'échelle de 160 000 000 ou moins. La règle trace un cercle (**ligne 8**), coloré en rouge (#CC3300 en **lige 10**) avec une taille de 12 pixels (**ligne 13**).

La seconde règle (**lignes 17-32**) correspond aux échelles intermédiaires, lorsque la vue est "zoomée partiellement". Les seuils d'échelle sont réglés en **lignes 19-20**, pour que la règle s'applique à toute carte avec un dénominateur d'échelle compris entre 160 000 000 and 320 000 000. (Le `<MinScaleDenominator>` est inclusif et le `<MaxScaleDenominator>` est exclusif pour qu'un niveau de zoom d'exactement 320 000 000 ne s'applique pas ici.) Mis à part l'échelle, la seule différence entre cette règle et la première est la taille du symbole, qui est à 8 pixels en **ligne 29**.

La troisième règle (**lignes 32-47**) est le dénominateur d'échelle le plus grand, correspondant à une carte avec "zoom arrière". Le seuil d'échelle est réglé en **line 35** pour que la règle s'applique à toute carte avec un dénominateur d'échelle supérieur ou égal à 320 000 000. A nouveau, la seule autre différence entre cette règle et les autres est la taille du symbole, qui est ici à 4 pixels en **ligne 44**.

Ce style a pour résultat des points représentés plus grands lorsque l'on zoome, et plus petits lorsque l'on effectue un zoom arrière.

### Lines

Bien que les lignes n'ayant qu'une longueur sans épaisseur semblent être des formes simples, il existe beaucoup d'options et de trucs les représenter de façon satisfaisante.

## Avis

Pour rester concis, les exemples de code présentés sur cette page ne sont **pas le code SLD complet** car ils omettent les informations SLD de début et de fin. Utilisez les liens pour télécharger les SLD complets de chaque exemple.

### Exemple de couche ligne

La couche ligne utilisée dans les exemples ci-dessous contient les données routières d'un pays fictif. Pour référence, le tableau des attributs des lignes de cette couche est inclus ci-dessous.

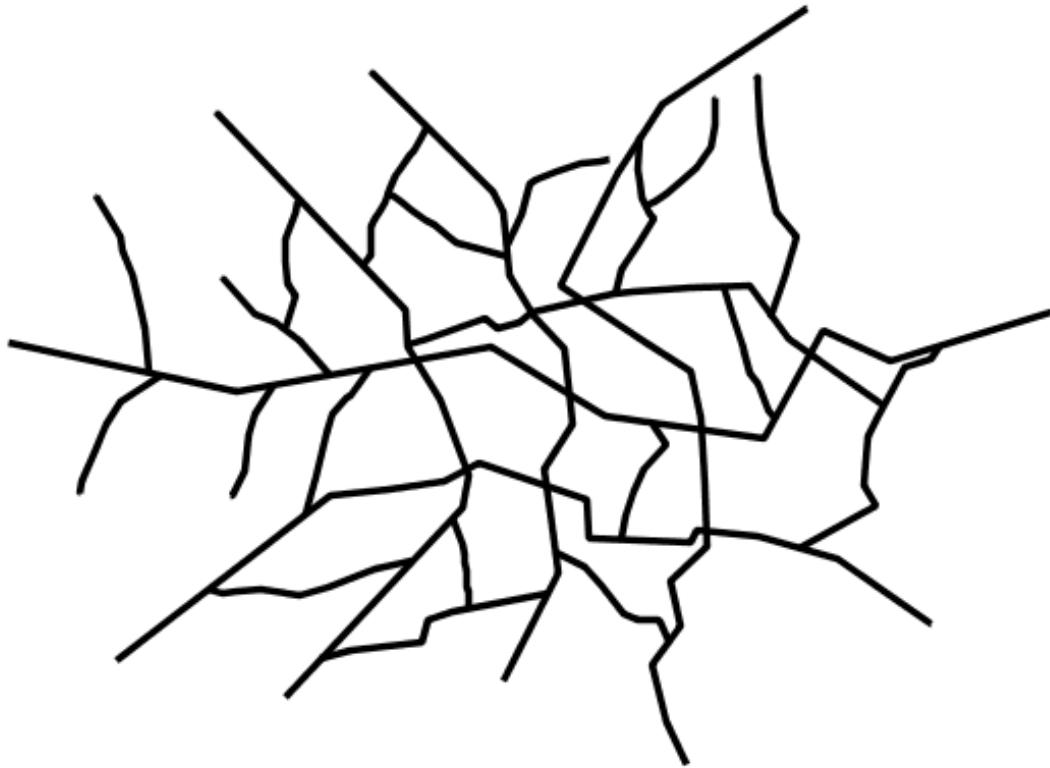
<b>fid</b> (Feature ID)	<b>name</b> (Road name)	<b>type</b> (Road class)
line.1	Latway	highway
line.2	Crescent Avenue	secondary
line.3	Forest Avenue	secondary
line.4	Longway	highway
line.5	Saxer Avenue	secondary
line.6	Ridge Avenue	secondary
line.7	Holly Lane	local-road
line.8	Mulberry Street	local-road
line.9	Nathan Lane	local-road
line.10	Central Street	local-road
line.11	Lois Lane	local-road
line.12	Rocky Road	local-road
line.13	Fleet Street	local-road
line.14	Diane Court	local-road
line.15	Cedar Trail	local-road
line.16	Victory Road	local-road
line.17	Highland Road	local-road
line.18	Easy Street	local-road
line.19	Hill Street	local-road
line.20	Country Road	local-road
line.21	Main Street	local-road
line.22	Jani Lane	local-road
line.23	Shinbone Alley	local-road
line.24	State Street	local-road
line.25	River Road	local-road

Téléchargez le shapefile

Une réponse efficace à la directive INSPIRE

### Ligne simple

Cet exemple précise que les lignes sont colorées en noir avec une épaisseur de 3 pixels.



*Ligne simple*

### Code

Consultez et téléchargez le SLD complet "Ligne simple"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#000000</CssParameter>
6          <CssParameter name="stroke-width">3</CssParameter>
7        </Stroke>
8      </LineSymbolizer>
9    </Rule>
10   </FeatureTypeStyle>
```

### Détails

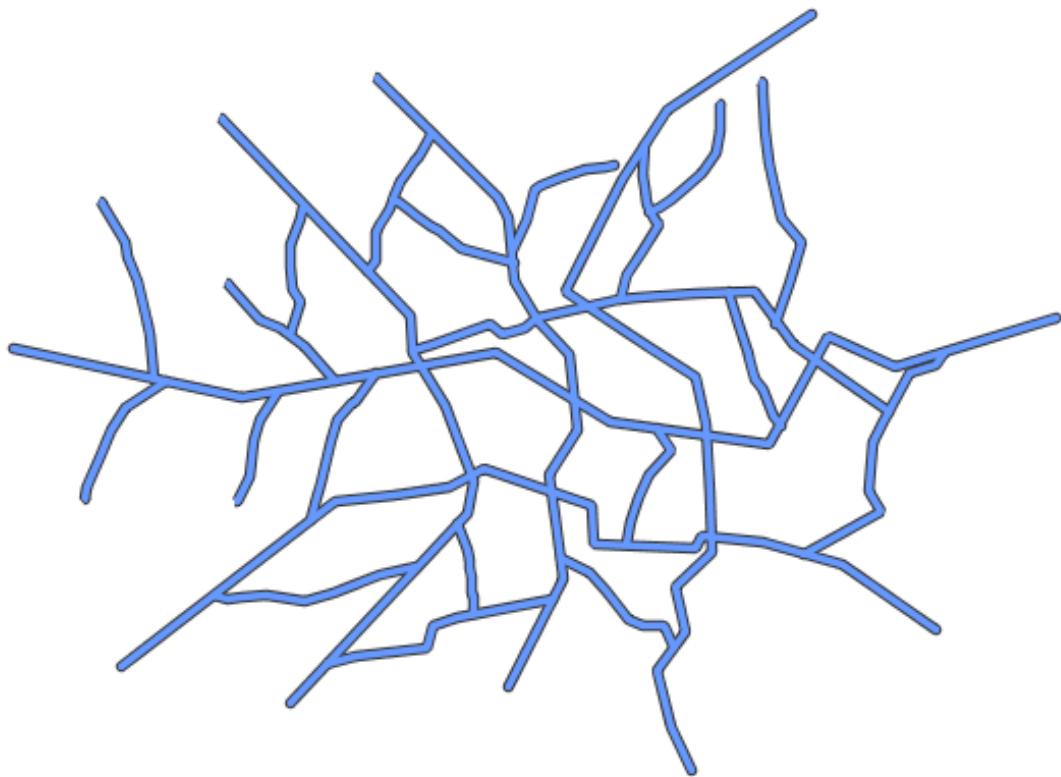
Dans ce SLD, il y a un `<Rule>` dans un `<FeatureTypeStyle>`, ce qui est la situation la plus simple possible. (Tous les exemples suivants contiendront un `<Rule>` et un `<FeatureTypeStyle>`, sauf mention contraire.) Symboliser les lignes se fait avec `<LineSymbolizer>` (**lignes 3-8**). La **ligne 5** règle la couleur des lignes à noir (#000000), et

Une réponse efficace à la directive INSPIRE

la **ligne 6** règle la largeur des lignes à 3 pixels.

### **Ligne avec bord**

Cet exemple dessine des lignes avec un remplissage de 3 pixels et un trait gris de 1 pixel.



*Ligne avec bord*

### **Code**

Consultez et téléchargez le SLD complet "Ligne avec bord"

```
1  <FeatureTypeStyle>
2      <Rule>
3          <LineSymbolizer>
4              <Stroke>
5                  <CssParameter name="stroke">#333333</CssParameter>
6                  <CssParameter name="stroke-width">5</CssParameter>
7                  <CssParameter name="stroke-linecap">round</CssParameter>
8              </Stroke>
9          </LineSymbolizer>
10     </Rule>
11 </FeatureTypeStyle>
12 <FeatureTypeStyle>
13     <Rule>
14         <LineSymbolizer>
```

```
15      <Stroke>
16          <CssParameter name="stroke">#6699FF</CssParameter>
17          <CssParameter name="stroke-width">3</CssParameter>
18          <CssParameter name="stroke-linecap">round</CssParameter>
19      </Stroke>
20  </LineSymbolizer>
21 </Rule>
22 </FeatureTypeStyle>
```

### Détails

En SLD, les lignes n'ont pas de notion de "remplissage", mais seulement "trait". A la différence des polygones, il n'est pas possible de symboliser les "bords" d'une géométrie ligne. Il est cependant possible d'obtenir cet effet en traçant chaque ligne deux fois : une fois avec une certaine largeur, une nouvelle fois avec une largeur un peu plus faible. Ceci donne l'illusion d'un remplissage avec bords en recouvrant partout les lignes épaisses, sauf le long des bords des lignes plus fines.

Comme chaque ligne est dessinée deux fois, l'ordre de tracé est très important. Dans ce style, toutes les lignes grises sont dessinées d'abord avec le premier `<FeatureTypeStyle>`, et sont suivies par toutes les lignes bleues du second `<FeatureTypeStyle>`. GeoServer va tracer tous les `<FeatureTypeStyle>` dans l'ordre d'apparition dans le SLD. Ceci permet non seulement de s'assurer que les lignes bleues ne seront pas recouvertes par les lignes grises, mais permet aussi un dessin propre aux intersections, à l'endroit où les lignes bleues se "connectent".

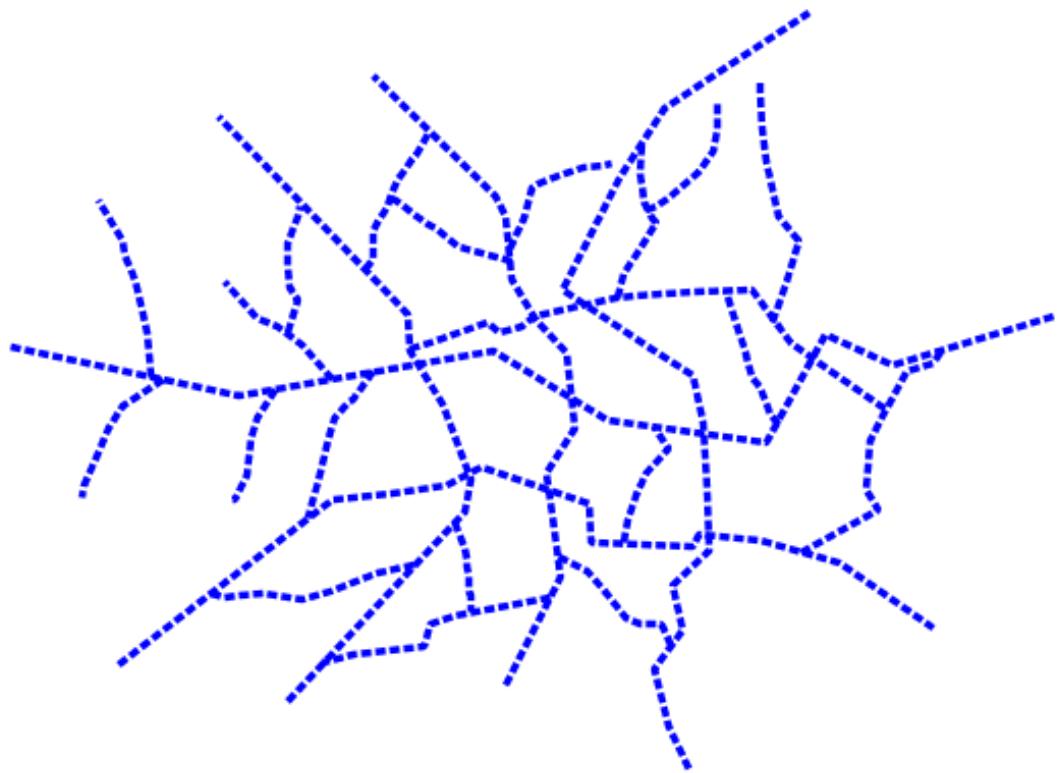
Dans cet exemple, les **lignes 1-11** comprennent le premier `<FeatureTypeStyle>`, qui est la ligne extérieure (le "trait"). La **ligne 5** règle la couleur de la ligne à gris foncé (#333333), la **ligne 6** règle la largeur de cette ligne à 5 pixels, et la **ligne 7** arrondit les extrémités de la ligne, plats par défaut. (Lorsque l'on travaille avec des lignes avec bords, utiliser le paramètre `stroke-linecap` permet de s'assurer que les fins de lignes ont des bords correctement dessinés.)

Les **lignes 12-22** comprennent le second `<FeatureTypeStyle>` qui correspond à l'intérieur de la ligne (le "remplissage"). La **Ligne 16** règle la couleur de la ligne à bleu moyen (#6699FF), la **ligne 17** règle la largeur de cette ligne à 3 pixels, et la **ligne 18** arrondit les extrémités des lignes au lieu de les dessiner plates.

Le résultat est une ligne bleue de 3 pixels avec un bord d'un pixel, puisque la ligne grise de 5 pixels affichera 1 pixel de part et d'autre de la ligne bleue de 3 pixels.

### Ligne pointillée

Cet exemple modifie *Ligne simple* pour créer une ligne pointillée constituée de segments de 5 pixels alternés avec des espaces de 2 pixels.



*Ligne pointillée*

### **Code**

Consultez et téléchargez le SLD complet "ligne pointillée"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#0000FF</CssParameter>
6          <CssParameter name="stroke-width">3</CssParameter>
7          <CssParameter name="stroke-dasharray">5 2</CssParameter>
8        </Stroke>
9      </LineSymbolizer>
10    </Rule>
11  </FeatureTypeStyle>
```

### **Détails**

Dans cet exemple, la **ligne 5** règle la couleur des lignes à bleu (#0000FF) et la **ligne 6** règle la largeur des lignes à 3 pixels. La **ligne 7** détermine la composition des tirets. la valeur 5 2 crée un motif de 5 pixels de ligne dessinée, suivi de 2 pixels sans ligne.

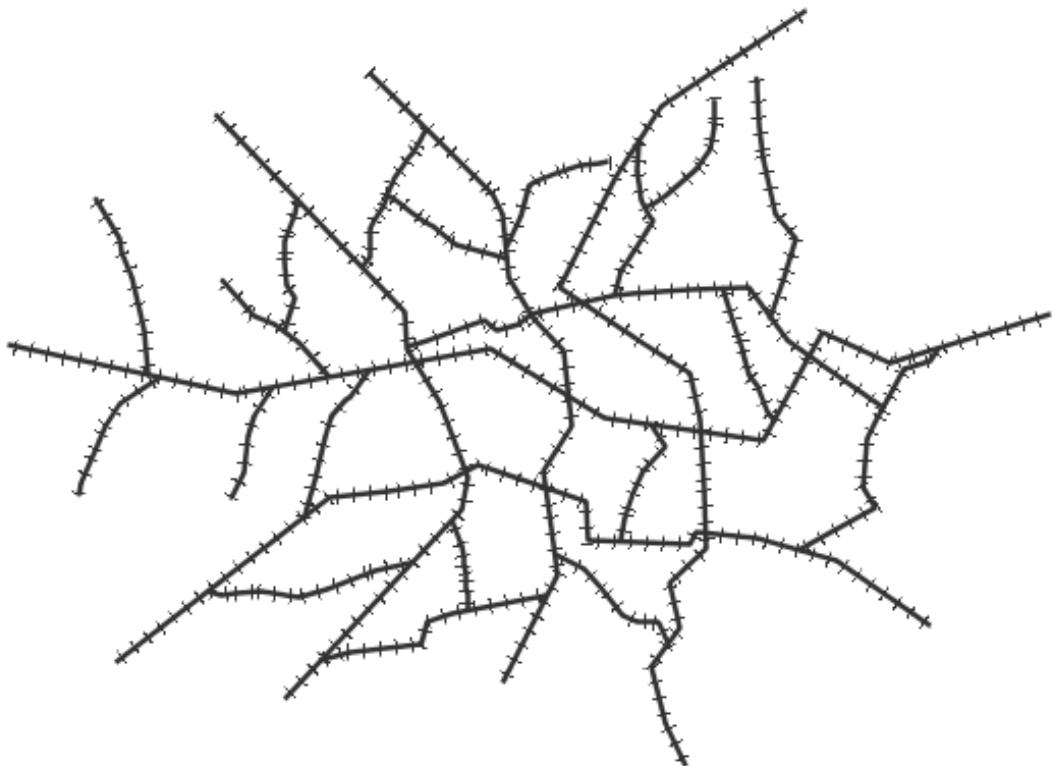
### **Voie ferrée (hachures)**

## Une réponse efficace à la directive INSPIRE

Cet exemple utilise des hachures pour fabriquer un style voie ferrée. Ligne et hachures sont en noir, avec une épaisseur de 2 pixels pour la ligne principale et de 1 pixel pour les hachures perpendiculaires.

### Note

Cet exemple utilise une extension SLD de GeoServer. Les hachures ne font pas partie de la spécification SLD 1.0.



*Voie ferrée (hachures)*

### Code

Consultez et téléchargez le SLD complet "Voie ferrée (hachures)"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#333333</CssParameter>
6          <CssParameter name="stroke-width">3</CssParameter>
7        </Stroke>
8      </LineSymbolizer>
```

```
9      </Rule>
10     <Rule>
11       <LineSymbolizer>
12         <Stroke>
13           <GraphicStroke>
14             <Graphic>
15               <Mark>
16                 <WellKnownName>shape://vertline</WellKnownName>
17                 <Stroke>
18                   <CssParameter name="stroke">#333333</CssParameter>
19                   <CssParameter name="stroke-width">1</CssParameter>
20                 </Stroke>
21               </Mark>
22               <Size>12</Size>
23             </Graphic>
24           </GraphicStroke>
25         </Stroke>
26       </LineSymbolizer>
27     </Rule>
28   </FeatureTypeStyle>
```

### Détails

Dans cet exemple il y a deux règles, chacune contenant un `<LineSymbolizer>` (Chaque `<LineSymbolizer>` doit être dans une règle distincte). La première règle, en **lignes 2-8**, trace une ligne standard, avec la **ligne 5** dessinant les lignes en gris foncé (#333333) et la **ligne 6** paramétrant la largeur des lignes à 2 pixels.

Les hachures sont invoquées dans la deuxième règle, en **lignes 10-27**. La **ligne 16** dit que la règle trace des hachures verticales (`shape://vertline`) perpendiculaires à la géométrie de la ligne. Les **lignes 18-19** règlent la couleur des hachures à gris foncé (#333333) et la largeur à 1 pixel. Enfin, la ligne 22 dit que la longueur des hachures et la distance entre chaque hachure doivent être de 12 pixels.

### Symboles graphiques espacés

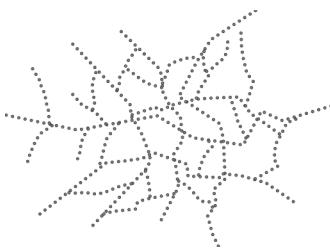
Cet exemple utilise un symbole graphique "trait" et une matrice de hachures pour créer un type de ligne "point espace". En l'absence de matrice de hachures, les lignes seraient remplies de points successifs se touchant.

L'ajout de la matrice de hachures permet de contrôler la taille de l'espace entre un symbole et le suivant.

### Note

Cet exemple ne fonctionnera probablement pas avec des systèmes tiers supportant SLD. Bien que ce soit parfaitement conforme à SLD, à notre connaissance aucun autre système ne permet de combiner l'utilisation de dasharray et de symboles graphiques "trait" (la spécification SLD ne dit pas ce que cette combinaison est censée produire).

Une réponse efficace à la directive INSPIRE



*Symboles espacés le long d'une ligne*

## Code

Consultez et téléchargez le SLD complet "Symboles espacés"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <GraphicStroke>
6            <Graphic>
7              <Mark>
8                <WellKnownName>circle</WellKnownName>
9                <Stroke>
10               <CssParameter name="stroke">#333333</CssParameter>
11               <CssParameter name="stroke-width">1</CssParameter>
12             </Stroke>
13             <Fill>
14               <CssParameter name="stroke">#666666</CssParameter>
15             </Fill>
16           </Mark>
17           <Size>4</Size>
18           <CssParameter name="stroke-dasharray">4 6</CssParameter>
19         </Graphic>
20       </GraphicStroke>
21     </Stroke>
22   </LineSymbolizer>
23 </Rule>
24 </FeatureTypeStyle>
```

## Détails

Cet exemple, comme les précédents, utilise un `GraphicStroke` pour placer un symbole graphique le long d'une ligne. Le symbole défini en **lignes 7-16** est un cercle gris de 4 pixels avec un contour gris foncé. L'espacement entre les symboles est contrôlé par `dasharray` en **ligne 18**, réglant 4 pixels de tracé, juste assez pour dessiner le cercle, puis 6 pixels crayon levé, ce qui produit un espace.

## Alternner les symboles avec `dashoffset`

Cet exemple montre comment créer un style de ligne complexe qui alterne un symbole et un segment de ligne. L'exemple repose sur les connaissances acquises dans les sections précédentes:

- `dasharray` permet de contrôler le comportement baisser crayon/lever crayon et génère des lignes pointillées

## Une réponse efficace à la directive INSPIRE

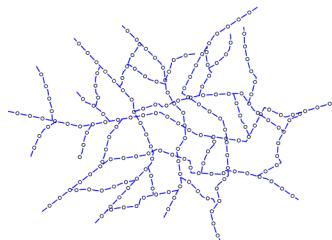
- *GraphicStroke* permet de placer des symboles le long d'une ligne
- en combinant les deux, il est possible de contrôler l'espacement entre symboles

Cet exemple ajoute l'utilisation du décalage *dashoffset* qui contrôle à quel endroit de la séquence *dasharray* le moteur commence à dessiner le motif répété. Par exemple, si l'on a une matrice de hachures de 5 10` et un décalage de ``7 , le moteur débutera à tracer le motif répété 7 pixels après le début, en sautant la section "5 pixels crayon baissé" et 3 pixels supplémentaires de la section "crayon levé", traçant le reste des 8 pixels crayon levé, puis 5 baissé, 10 levé, et ainsi de suite.

Ceci peut être utilisé pour créer deux séquences de matrices de hachures synchronisées, l'une dessinant le long d'une ligne des segments, l'autre des symboles, comme dans l'exemple suivant.

### Note

Cet exemple ne fonctionnera probablement pas avec des systèmes tiers supportant SLD. Bien que ce soit parfaitement conforme à SLD, à notre connaissance aucun autre système ne permet de combiner l'utilisation de *dasharray* et de symboles graphiques "trait" (la spécification SLD ne dit pas ce que cette combinaison est censée produire).



Hachures et symboles

### Code

Consultez et téléchargez le SLD complet "Symboles espacés"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#0000FF</CssParameter>
6          <CssParameter name="stroke-width">1</CssParameter>
7          <CssParameter name="stroke-dasharray">10 10</CssParameter>
8        </Stroke>
9      </LineSymbolizer>
10     <LineSymbolizer>
11       <Stroke>
12         <GraphicStroke>
13           <Graphic>
14             <Mark>
15               <WellKnownName>circle</WellKnownName>
16               <Stroke>
17                 <CssParameter name="stroke">#000033</CssParameter>
```

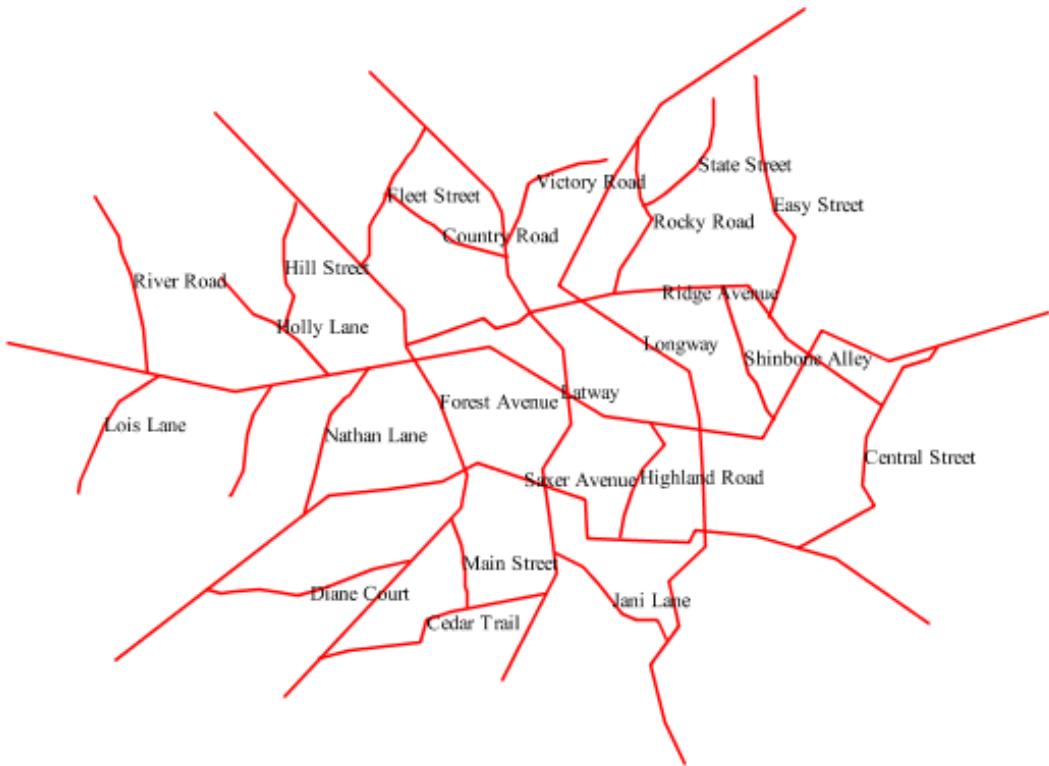
```
18      <CssParameter name="stroke-width">1</CssParameter>
19      </Stroke>
20  </Mark>
21      <Size>5</Size>
22      <CssParameter name="stroke-dasharray">5 15</CssParameter>
23      <CssParameter name="stroke-dashoffset">7.5</CssParameter>
24      </Graphic>
25      </GraphicStroke>
26  </Stroke>
27  </LineSymbolizer>
28  </Rule>
29  </FeatureTypeStyle>
```

### Détails

Dans cet exemple, deux styles de ligne basés sur des matrices de hachure sont utilisés pour générer une séquence alternée. Le premier, défini en **lignes 3-9**, est une simple matrice de hachure sur ligne alternant 10 pixels crayon baissé et 10 pixels crayon levé. Le second, défini en **lignes 10-27**, alterne des cercles vides de 5 pixels et des espaces de 15 pixels. Pour que les deux styles soient alternés, le second utilise un décalage de 7.5, faisant démarrer la séquence avec 12.5 pixels d'espace, suivis d'un cercle (qui est alors centré entre les deux segments de ligne de l'autre motif), puis 15 pixels d'espace, et ainsi de suite.

### **Line with default label**

This example shows a text label on the simple line. This is how a label will be displayed in the absence of any other customization.



*Line with default label*

### **Code**

View and download the full "Line with default label" SLD

```
1  <FeatureTypeStyle>
2      <Rule>
3          <LineSymbolizer>
4              <Stroke>
5                  <CssParameter name="stroke">#FF0000</CssParameter>
6              </Stroke>
7          </LineSymbolizer>
8          <TextSymbolizer>
9              <Label>
10                 <ogc:PropertyName>name</ogc:PropertyName>
11             </Label>
12             <Fill>
13                 <CssParameter name="fill">#000000</CssParameter>
14             </Fill>
15         </TextSymbolizer>
16     </Rule>
17 </FeatureTypeStyle>
```

### **Details**

Une réponse efficace à la directive INSPIRE

In this example, there is one rule with a `<LineSymbolizer>` and a `<TextSymbolizer>`. The `<LineSymbolizer>` (**lines 3-7**) draws red lines (#FF0000). Since no width is specified, the default is set to 1 pixel. The `<TextSymbolizer>` (**lines 8-15**) determines the labeling of the lines. **Lines 9-11** specify that the text of the label will be determined by the value of the "name" attribute for each line. (Refer to the attribute table in the [Exemple de couche ligne](#) section if necessary.) **Line 13** sets the text color to black. All other details about the label are set to the renderer default, which here is Times New Roman font, font color black, and font size of 10 pixels.

### **Label following line**

This example renders the text label to follow the contour of the lines.

#### **Note**

Labels following lines is an SLD extension specific to GeoServer. It is not part of the SLD 1.0 specification.



*Label following line*

### **Code**

View and download the full "Label following line" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#FF0000</CssParameter>
6        </Stroke>
7      </LineSymbolizer>
8      <TextSymbolizer>
9        <Label>
10       <ogc:PropertyName>name</ogc:PropertyName>
11     </Label>
12     <Fill>
13       <CssParameter name="fill">#000000</CssParameter>
14     </Fill>
15     <VendorOption name="followLine">true</VendorOption>
16     <LabelPlacement>
17       <LinePlacement />
18     </LabelPlacement>
19   </TextSymbolizer>
20 </Rule>
21 </FeatureTypeStyle>
```

### Details

As the [Alterner les symboles avec dashoffset](#) example showed, the default label behavior isn't very optimal. The label is displayed at a tangent to the line itself, leading to uncertainty as to which label corresponds to which line.

This example is similar to the [Alterner les symboles avec dashoffset](#) example with the exception of **lines 15-18**. **Line 15** sets the option to have the label follow the line, while **lines 16-18** specify that the label is placed along a line. If `<LinePlacement />` is not specified in an SLD, then `<PointPlacement />` is assumed, which isn't compatible with line-specific rendering options.

### Note

Not all labels are shown due to label conflict resolution. See the next section on [Optimized label placement](#) for an example of how to maximize label display.

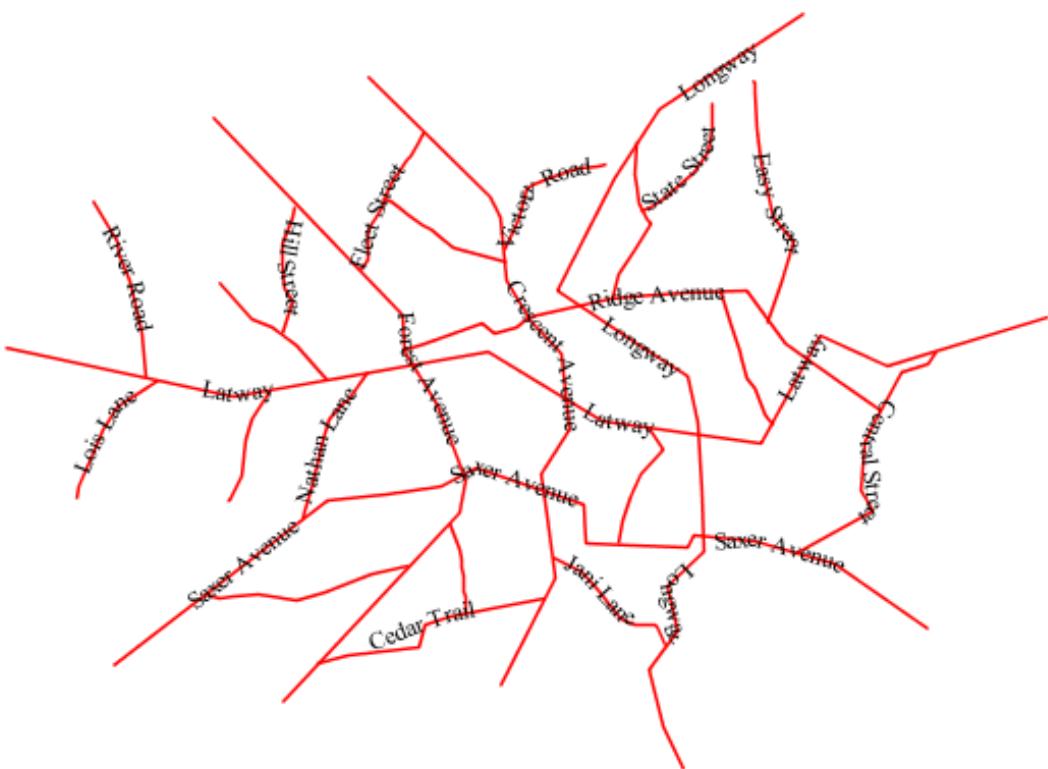
### Optimized label placement

This example optimizes label placement for lines such that the maximum number of labels are displayed.

### Note

This example uses options that are specific to GeoServer and are not part of the SLD 1.0 specification.

## Une réponse efficace à la directive INSPIRE



*Optimized label*

Code

[View and download the full "Optimized label" SLD](#)

```
1 <FeatureTypeStyle>
2   <Rule>
3     <LineSymbolizer>
4       <Stroke>
5         <CssParameter name="stroke">#FF0000</CssParameter>
6       </Stroke>
7     </LineSymbolizer>
8     <TextSymbolizer>
9       <Label>
10      <ogc:PropertyName>name</ogc:PropertyName>
11    </Label>
12    <Fill>
13      <CssParameter name="fill">#000000</CssParameter>
14    </Fill>
15    <VendorOption name="followLine">true</VendorOption>
16    <VendorOption name="maxAngleDelta">90</VendorOption>
17    <VendorOption name="maxDisplacement">400</VendorOption>
18    <VendorOption name="repeat">150</VendorOption>
19    <LabelPlacement>
```

```
20      <LinePlacement />
21    </LabelPlacement>
22  </TextSymbolizer>
23  </Rule>
24 </FeatureTypeStyle>
```

### Details

GeoServer uses "conflict resolution" to ensure that labels aren't drawn on top of other labels, obscuring them both. This accounts for the reason why many lines don't have labels in the previous example, [Label following line](#). While this setting can be toggled, it is usually a good idea to leave it on and use other label placement options to ensure that labels are drawn as often as desired and in the correct places. This example does just that.

This example is similar to the previous example, [Label following line](#). The only differences are contained in **lines 16-18**. **Line 16** sets the maximum angle that the label will follow. This sets the label to never bend more than 90 degrees to prevent the label from becoming illegible due to a pronounced curve or angle. **Line 17** sets the maximum displacement of the label to be 400 pixels. In order to resolve conflicts with overlapping labels, GeoServer will attempt to move the labels such that they are no longer overlapping. This value sets how far the label can be moved relative to its original placement. Finally, **line 18** sets the labels to be repeated every 150 pixels. A feature will typically receive only one label, but this can cause confusion for long lines. Setting the label to repeat ensures that the line is always labeled locally.

### Optimized and styled label

This example improves the style of the labels from the [Optimized label placement](#) example.



*Optimized and styled label*

### Code

View and download the full "Optimized and styled label" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <LineSymbolizer>
4        <Stroke>
5          <CssParameter name="stroke">#FF0000</CssParameter>
6        </Stroke>
7      </LineSymbolizer>
8      <TextSymbolizer>
9        <Label>
10       <ogc:PropertyName>name</ogc:PropertyName>
11     </Label>
12     <Fill>
13       <CssParameter name="fill">#000000</CssParameter>
14     </Fill>
15     <Font>
16       <CssParameter name="font-family">Arial</CssParameter>
17       <CssParameter name="font-size">10</CssParameter>
18       <CssParameter name="font-style">normal</CssParameter>
19       <CssParameter name="font-weight">bold</CssParameter>
```

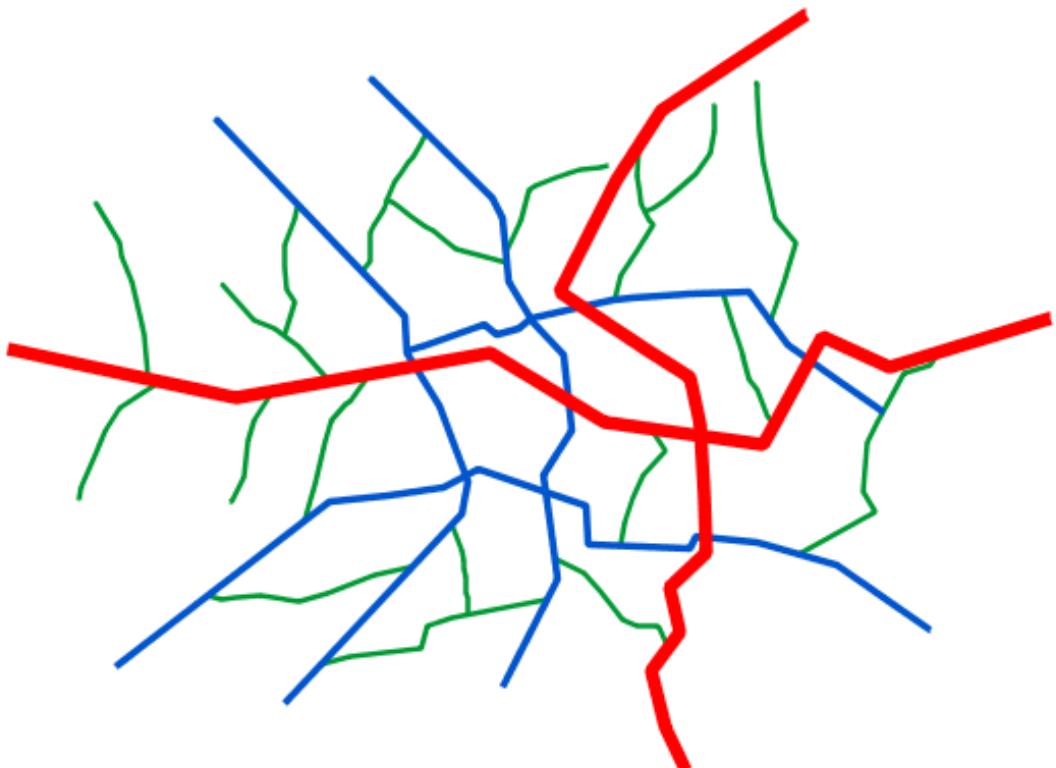
```
20      </Font>
21      <VendorOption name="followLine">true</VendorOption>
22      <VendorOption name="maxAngleDelta">90</VendorOption>
23      <VendorOption name="maxDisplacement">400</VendorOption>
24      <VendorOption name="repeat">150</VendorOption>
25      <LabelPlacement>
26          <LinePlacement />
27      </LabelPlacement>
28  </TextSymbolizer>
29 </Rule>
30 </FeatureTypeStyle>
```

### Details

This example is similar to the [Optimized label placement](#). The only difference is in the font information, which is contained in **lines 15-20**. **Line 16** sets the font family to be "Arial", **line 17** sets the font size to 10, **line 18** sets the font style to "normal" (as opposed to "italic" or "oblique"), and **line 19** sets the font weight to "bold" (as opposed to "normal").

### Attribute-based line

This example styles the lines differently based on the "type" (Road class) attribute.



Attribute-based line

### Code

## Une réponse efficace à la directive INSPIRE

View and download the full "Attribute-based line" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <Name>local-road</Name>
4      <ogc:Filter>
5        <ogc:PropertyIsEqualTo>
6          <ogc:PropertyName>type</ogc:PropertyName>
7          <ogc:Literal>local-road</ogc:Literal>
8        </ogc:PropertyIsEqualTo>
9      </ogc:Filter>
10     <LineSymbolizer>
11       <Stroke>
12         <CssParameter name="stroke">#009933</CssParameter>
13         <CssParameter name="stroke-width">2</CssParameter>
14       </Stroke>
15     </LineSymbolizer>
16   </Rule>
17 </FeatureTypeStyle>
18 <FeatureTypeStyle>
19   <Rule>
20     <Name>secondary</Name>
21     <ogc:Filter>
22       <ogc:PropertyIsEqualTo>
23         <ogc:PropertyName>type</ogc:PropertyName>
24         <ogc:Literal>secondary</ogc:Literal>
25       </ogc:PropertyIsEqualTo>
26     </ogc:Filter>
27     <LineSymbolizer>
28       <Stroke>
29         <CssParameter name="stroke">#0055CC</CssParameter>
30         <CssParameter name="stroke-width">3</CssParameter>
31       </Stroke>
32     </LineSymbolizer>
33   </Rule>
34 </FeatureTypeStyle>
35 <FeatureTypeStyle>
36   <Rule>
37     <Name>highway</Name>
38     <ogc:Filter>
39       <ogc:PropertyIsEqualTo>
40         <ogc:PropertyName>type</ogc:PropertyName>
41         <ogc:Literal>highway</ogc:Literal>
42       </ogc:PropertyIsEqualTo>
43     </ogc:Filter>
44     <LineSymbolizer>
45       <Stroke>
46         <CssParameter name="stroke">#FF0000</CssParameter>
47         <CssParameter name="stroke-width">6</CssParameter>
48       </Stroke>
49     </LineSymbolizer>
50   </Rule>
51 </FeatureTypeStyle>
```

## Details

### Note

Refer to the [Exemple de couche ligne](#) to see the attributes for the layer. This example has eschewed labels in order to simplify the style, but you can refer to the example [Optimized and styled label](#) to see which attributes correspond to which points.

There are three types of road classes in our fictional country, ranging from back roads to high-speed freeways: "highway", "secondary", and "local-road". In order to handle each case separately, there is more than one <FeatureTypeStyle>, each containing a single rule. This ensures that each road type is rendered in order, as each <FeatureTypeStyle> is drawn based on the order in which it appears in the SLD.

The three rules are designed as follows:

Rule order	Rule name / type	Color	Size
1	local-road	#009933 (green)	2
2	secondary	#0055CC (blue)	3
3	highway	#FF0000 (red)	6

**Lines 2-16** comprise the first <Rule>. **Lines 4-9** set the filter for this rule, such that the "type" attribute has a value of "local-road". If this condition is true for a particular line, the rule is rendered according to the <LineSymbolizer> which is on **lines 10-15**. **Lines 12-13** set the color of the line to be a dark green (#009933) and the width to be 2 pixels.

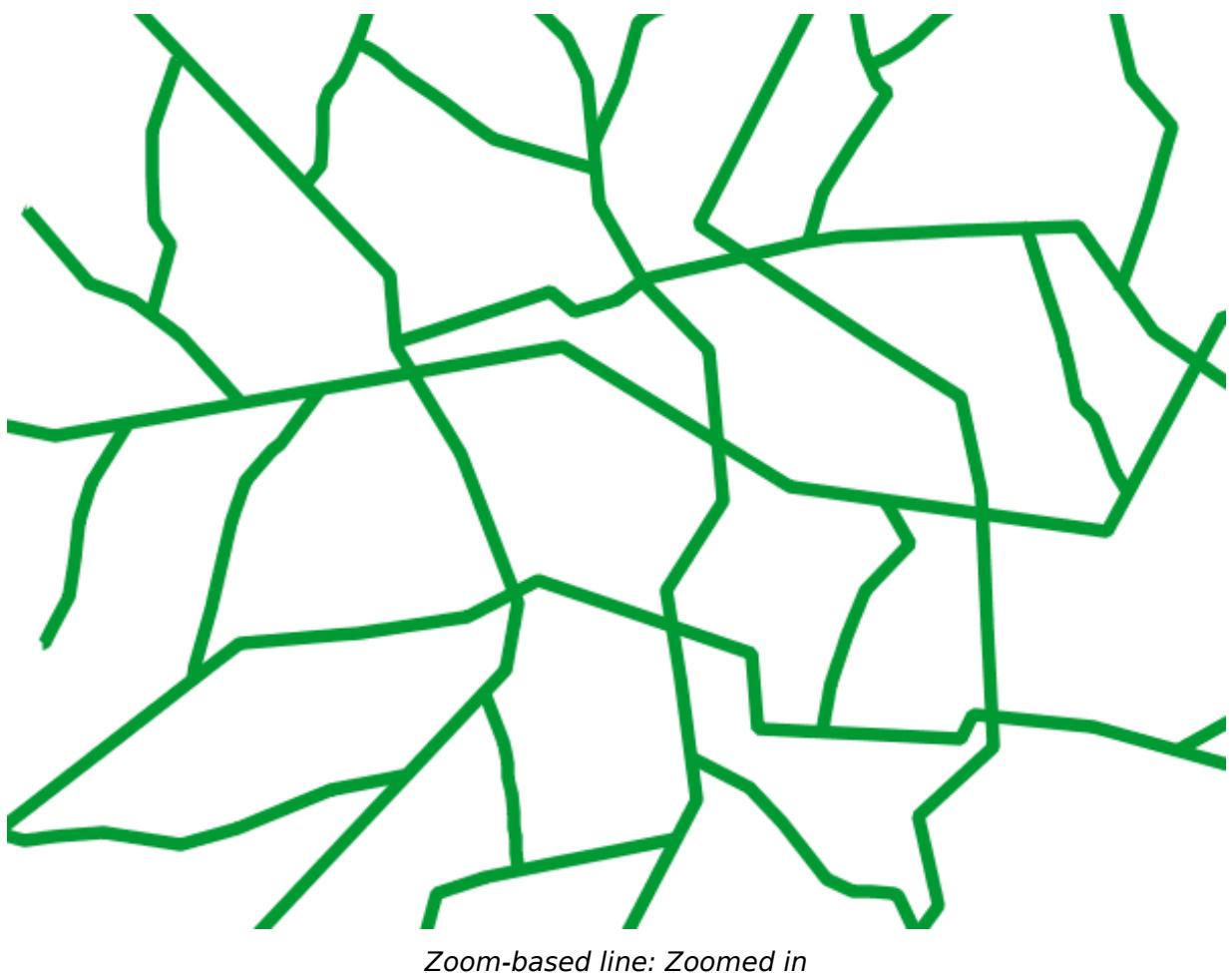
**Lines 19-33** comprise the second <Rule>. **Lines 21-26** set the filter for this rule, such that the "type" attribute has a value of "secondary". If this condition is true for a particular line, the rule is rendered according to the <LineSymbolizer> which is on **lines 27-32**. **Lines 29-30** set the color of the line to be a dark blue (#0055CC) and the width to be 3 pixels, making the lines slightly thicker than the "local-road" lines and also a different color.

**Lines 36-50** comprise the third and final <Rule>. **Lines 38-43** set the filter for this rule, such that the "type" attribute has a value of "primary". If this condition is true for a particular line, the rule is rendered according to the <LineSymbolizer> which is on **lines 44-49**. **Lines 46-47** set the color of the line to be a bright red (#FF0000) and the width to be 6 pixels, so that these lines are rendered on top of and thicker than the other two road classes. In this way, the "primary" roads are given priority in the map rendering.

### Zoom-based line

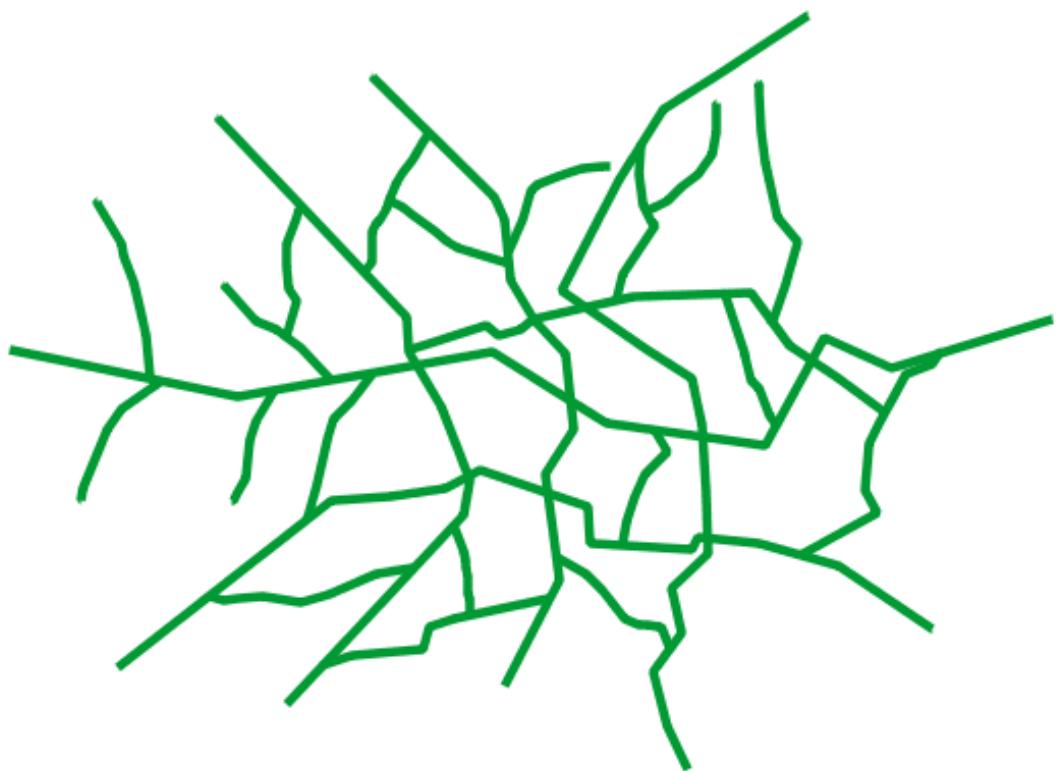
This example alters the [Ligne simple](#) style at different zoom levels.

Une réponse efficace à la directive INSPIRE

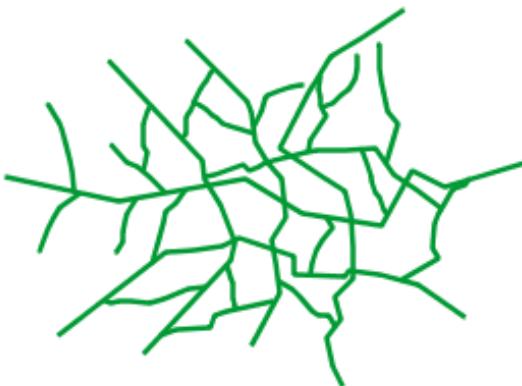


*Zoom-based line: Zoomed in*

Une réponse efficace à la directive INSPIRE



*Zoom-based line: Partially zoomed*



*Zoom-based line: Zoomed out*

### **Code**

View and download the full "Zoom-based line" SLD

```
1  <FeatureTypeStyle>
2      <Rule>
3          <Name>Large</Name>
4          <MaxScaleDenominator>180000000</MaxScaleDenominator>
5          <LineSymbolizer>
6              <Stroke>
7                  <CssParameter name="stroke">#009933</CssParameter>
8                  <CssParameter name="stroke-width">6</CssParameter>
9              </Stroke>
10         </LineSymbolizer>
11     </Rule>
12     <Rule>
13         <Name>Medium</Name>
14         <MinScaleDenominator>180000000</MinScaleDenominator>
15         <MaxScaleDenominator>360000000</MaxScaleDenominator>
16         <LineSymbolizer>
17             <Stroke>
18                 <CssParameter name="stroke">#009933</CssParameter>
19                 <CssParameter name="stroke-width">4</CssParameter>
```

```

20      </Stroke>
21  </LineSymbolizer>
22 </Rule>
23 <Rule>
24   <Name>Small</Name>
25   <MinScaleDenominator>360000000</MinScaleDenominator>
26   <LineSymbolizer>
27     <Stroke>
28       <CssParameter name="stroke">#009933</CssParameter>
29       <CssParameter name="stroke-width">2</CssParameter>
30     </Stroke>
31   </LineSymbolizer>
32 </Rule>
33 </FeatureTypeStyle>
```

### **Details**

It is often desirable to make shapes larger at higher zoom levels when creating a natural-looking map. This example varies the thickness of the lines according to the zoom level (or more accurately, scale denominator). Scale denominators refer to the scale of the map. A scale denominator of 10,000 means the map has a scale of 1:10,000 in the units of the map projection.

### **Note**

Determining the appropriate scale denominators (zoom levels) to use is beyond the scope of this example.

This style contains three rules. The three rules are designed as follows:

<b>Rule order</b>	<b>Rule name</b>	<b>Scale denominator</b>	<b>Line width</b>
1	Large	1:180,000,000 or less	6
2	Medium	1:180,000,000 to 1:360,000,000	4
3	Small	Greater than 1:360,000,000	2

The order of these rules does not matter since the scales denominated in each rule do not overlap.

The first rule (**lines 2-11**) is the smallest scale denominator, corresponding to when the view is "zoomed in". The scale rule is set on **line 4**, so that the rule will apply to any map with a scale denominator of 180,000,000 or less. **Line 7-8** draws the line to be dark green (#009933) with a width of 6 pixels.

The second rule (**lines 12-22**) is the intermediate scale denominator, corresponding to when the view is "partially zoomed". **Lines 14-15** set the scale such that the rule will apply to any map with scale denominators between 180,000,000 and 360,000,000. (The `<MinScaleDenominator>` is inclusive and the `<MaxScaleDenominator>` is exclusive, so a zoom level of exactly 360,000,000 would *not* apply here.) Aside from the scale, the only difference between this rule and the previous is the width of the lines, which is set to 4 pixels on **line 19**.

## Une réponse efficace à la directive INSPIRE

The third rule (**lines 23-32**) is the largest scale denominator, corresponding to when the map is "zoomed out". The scale rule is set on **line 25**, so that the rule will apply to any map with a scale denominator of 360,000,000 or greater. Again, the only other difference between this rule and the others is the width of the lines, which is set to 2 pixels on **line 29**.

The result of this style is that lines are drawn with larger widths as one zooms in and smaller widths as one zooms out.

## Polygons

Polygons are two dimensional shapes that contain both an outer edge (or "stroke") and an inside (or "fill"). A polygon can be thought of as an irregularly-shaped point and is styled in similar ways to points.

### Avis

The code examples shown on this page are **not the full SLD code**, as they omit the SLD header and footer information for the sake of brevity. Please use the links to download the full SLD for each example.

## Example polygons layer

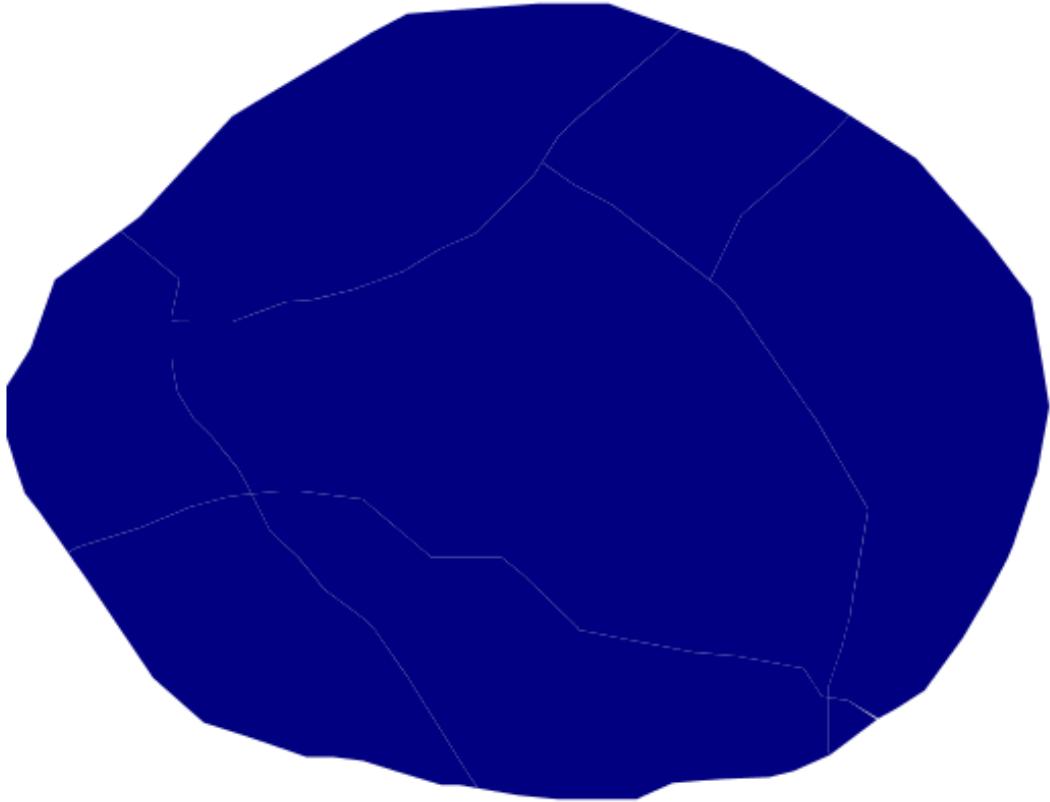
The polygons layer used below contains county information for a fictional country. For reference, the attribute table for the polygons is included below.

<b>fid</b> (Feature ID)	<b>name</b> (County name)	<b>pop</b> (Population)
polygon.1	Irony County	412234
polygon.2	Tracker County	235421
polygon.3	Dracula County	135022
polygon.4	Poly County	1567879
polygon.5	Bearing County	201989
polygon.6	Monte Cristo County	152734
polygon.7	Massive County	67123
polygon.8	Rhombus County	198029

Download the polygons shapefile

## Simple polygon

This example shows a polygon filled in blue.



*Simple polygon*

### **Code**

View and download the full "Simple polygon" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
5          <CssParameter name="fill">#000080</CssParameter>
6        </Fill>
7      </PolygonSymbolizer>
8    </Rule>
9  </FeatureTypeStyle>
```

### **Details**

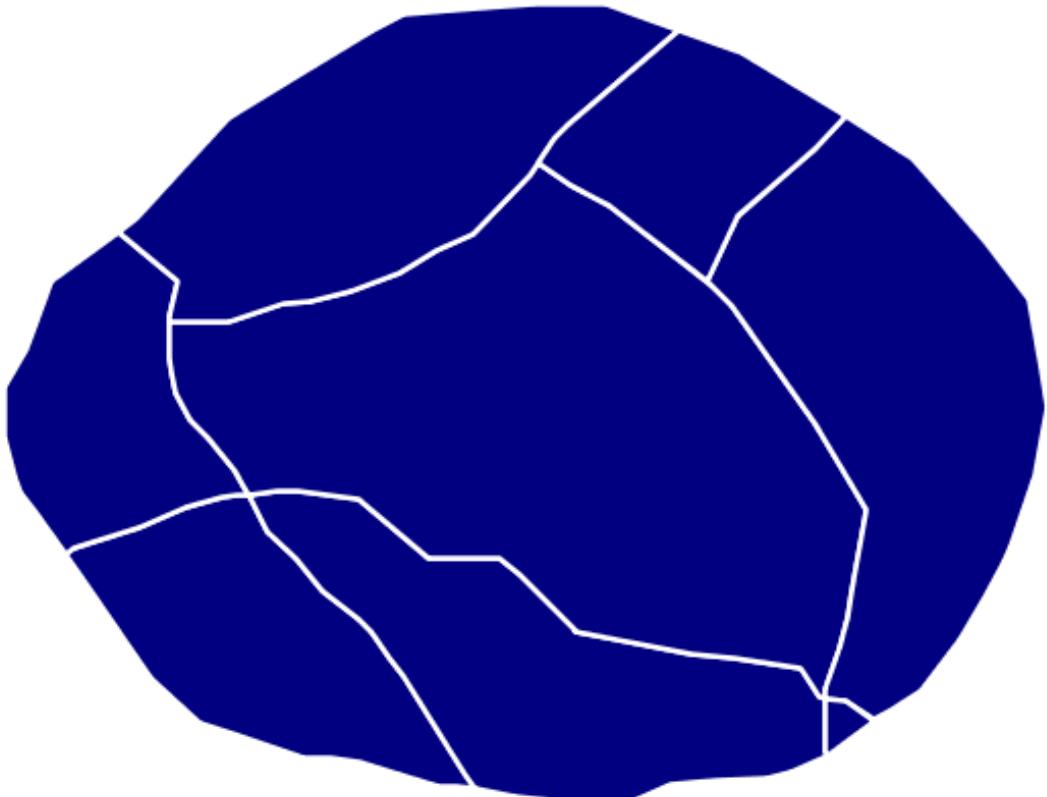
There is one `<Rule>` in one `<FeatureTypeStyle>` for this style, which is the simplest possible situation. (All subsequent examples will share this characteristic unless otherwise specified.) Styling polygons is accomplished via the `<PolygonSymbolizer>` (**lines 3-7**). **Line 5** specifies dark blue (#000080) as the polygon's fill color.

### Note

The light-colored borders around the polygons in the figure are artifacts of the renderer caused by the polygons being adjacent. There is no border in this style.

### **Simple polygon with stroke**

This example adds a 2 pixel white stroke to the [Simple polygon](#) example.



*Simple polygon with stroke*

### Code

View and download the full "Simple polygon with stroke" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
5          <CssParameter name="fill">#000080</CssParameter>
6        </Fill>
7        <Stroke>
8          <CssParameter name="stroke">#FFFFFF</CssParameter>
```

Une réponse efficace à la directive INSPIRE

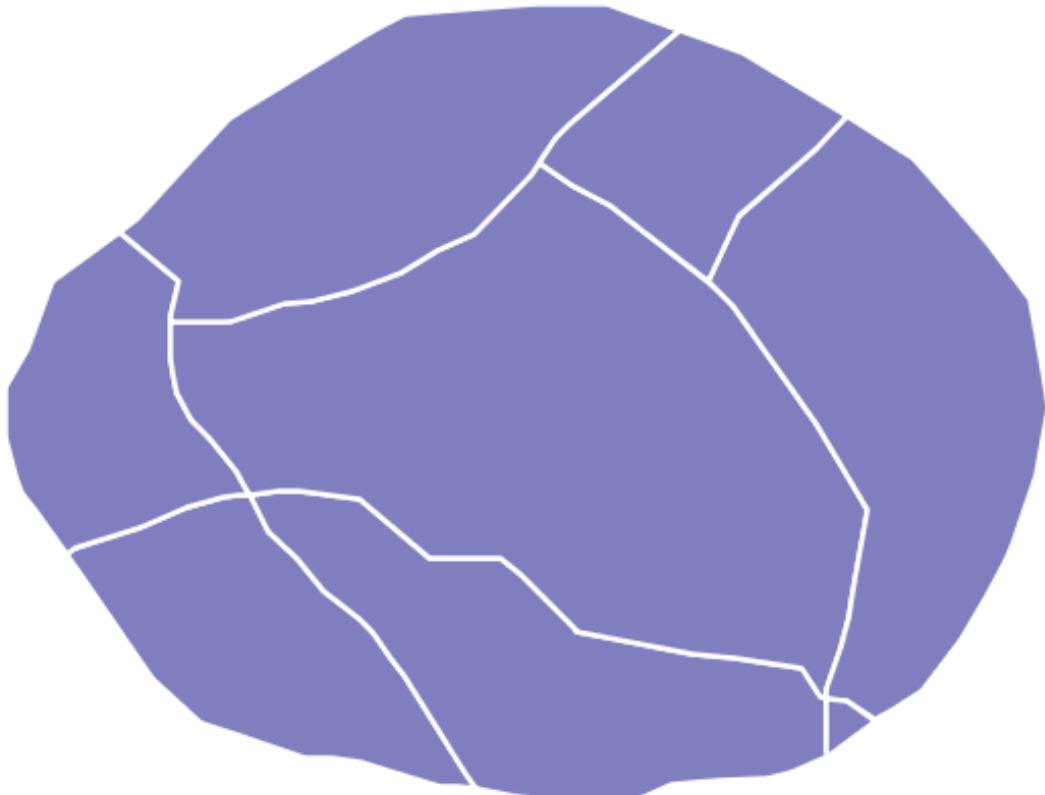
```
9      <CssParameter name="stroke-width">2</CssParameter>
10     </Stroke>
11   </PolygonSymbolizer>
12 </Rule>
13 </FeatureTypeStyle>
```

### Details

This example is similar to the [Simple polygon](#) example above, with the addition of the `<Stroke>` tag (**lines 7-10**). **Line 8** sets the color of stroke to white (#FFFFFF) and **line 9** sets the width of the stroke to 2 pixels.

### Transparent polygon

This example builds on the [Simple polygon with stroke](#) example and makes the fill partially transparent by setting the opacity to 50%.



*Transparent polygon*

### Code

View and download the full "Transparent polygon" SLD

```
1 <FeatureTypeStyle>
2   <Rule>
3     <PolygonSymbolizer>
```

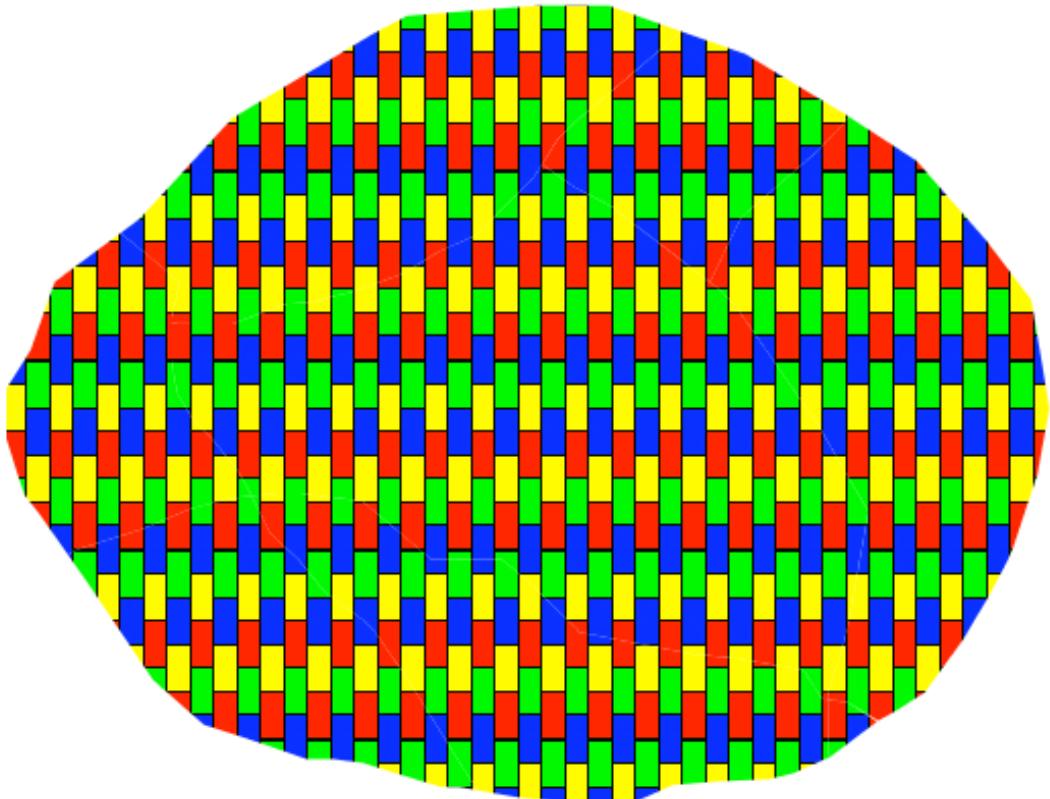
```
4      <Fill>
5          <CssParameter name="fill">#000080</CssParameter>
6          <CssParameter name="fill-opacity">0.5</CssParameter>
7      </Fill>
8      <Stroke>
9          <CssParameter name="stroke">#FFFFFF</CssParameter>
10         <CssParameter name="stroke-width">2</CssParameter>
11     </Stroke>
12   </PolygonSymbolizer>
13 </Rule>
14 </FeatureTypeStyle>
```

### Details

This example is similar to the [Simple polygon with stroke](#) example, save for defining the fill's opacity in **line 6**. The value of 0.5 results in partially transparent fill that is 50% opaque. An opacity value of 1 would draw the fill as 100% opaque, while an opacity value of 0 would result in a completely transparent (0% opaque) fill. In this example, since the background is white, the dark blue looks lighter. Were the points imposed on a dark background, the resulting color would be darker.

### Graphic fill

This example fills the polygons with a tiled graphic.



*Graphic fill*

## Code

View and download the full "Graphic fill" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
5          <GraphicFill>
6            <Graphic>
7              <ExternalGraphic>
8                <OnlineResource
9                  xlink:type="simple"
10                 xlink:href="colorblocks.png" />
11               <Format>image/png</Format>
12             </ExternalGraphic>
13             <Size>93</Size>
14           </Graphic>
15         </GraphicFill>
16       </Fill>
17     </PolygonSymbolizer>
18   </Rule>
19 </FeatureTypeStyle>
```

## Details

This style fills the polygon with a tiled graphic. This is known as an `<ExternalGraphic>` in SLD, to distinguish it from commonly-used shapes such as squares and circles that are "internal" to the renderer. **Lines 7-12** specify details for the graphic, with **line 10** setting the path and file name of the graphic and **line 11** indicating the file format (MIME type) of the graphic (image/png). Although a full URL could be specified if desired, no path information is necessary in **line 11** because this graphic is contained in the same directory as the SLD. **Line 13** determines the height of the displayed graphic in pixels; if the value differs from the height of the graphic then it will be scaled accordingly while preserving the aspect ratio.



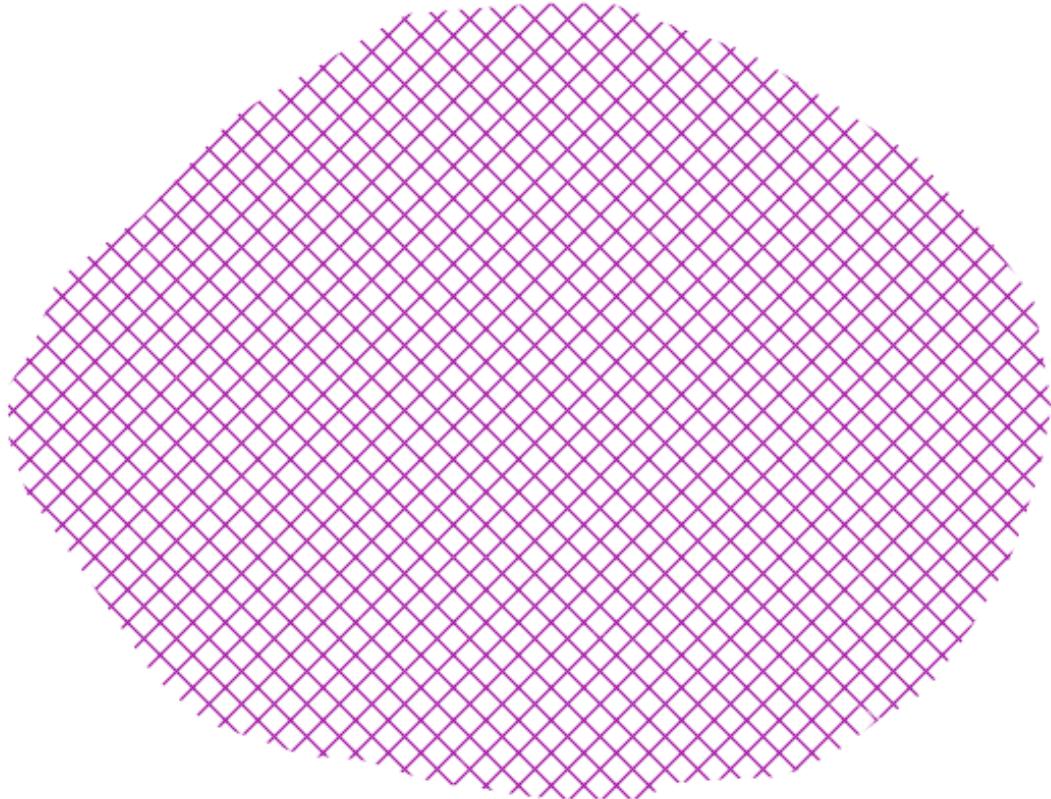
Graphic used for fill

## Hatching fill

This example fills the polygons with a hatching pattern.

### Note

This example leverages an SLD extension in GeoServer. Hatching is not part of the standard SLD 1.0 specification.



*Hatching fill*

### **Code**

View and download the full "Hatching fill" SLD

```
1   <FeatureTypeStyle>
2     <Rule>
3       <PolygonSymbolizer>
4         <Fill>
5           <GraphicFill>
6             <Graphic>
7               <Mark>
8                 <WellKnownName>shape://times</WellKnownName>
9                 <Stroke>
10                <CssParameter name="stroke">#990099</CssParameter>
11                <CssParameter name="stroke-width">1</CssParameter>
12                </Stroke>
13              </Mark>
14              <Size>16</Size>
15            </Graphic>
16          </GraphicFill>
17        </Fill>
18      </PolygonSymbolizer>
```

Une réponse efficace à la directive INSPIRE

```
19      </Rule>
20  </FeatureTypeStyle>
```

### Details

In this example, there is a `<GraphicFill>` tag as in the [Graphic fill](#) example, but a `<Mark>` (**lines 7-13**) is used instead of an `<ExternalGraphic>`. **Line 8** specifies a "times" symbol (an "x") be tiled throughout the polygon. **Line 10** sets the color to purple (#990099), **line 11** sets the width of the hatches to 1 pixel, and **line 14** sets the size of the tile to 16 pixels. Because hatch tiles are always square, the `<Size>` sets both the width and the height.

### Polygon with default label

This example shows a text label on the polygon. In the absence of any other customization, this is how a label will be displayed.



*Polygon with default label*

### Code

View and download the full "Polygon with default label" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
```

```
5      <CssParameter name="fill">#40FF40</CssParameter>
6    </Fill>
7    <Stroke>
8      <CssParameter name="stroke">#FFFFFF</CssParameter>
9      <CssParameter name="stroke-width">2</CssParameter>
10     </Stroke>
11   </PolygonSymbolizer>
12   <TextSymbolizer>
13     <Label>
14       <ogc:PropertyName>name</ogc:PropertyName>
15     </Label>
16   </TextSymbolizer>
17 </Rule>
18 </FeatureTypeStyle>
```

### Details

In this example there is a `<PolygonSymbolizer>` and a `<TextSymbolizer>`. **Lines 3-11** comprise the `<PolygonSymbolizer>`. The fill of the polygon is set on **line 5** to a light green (#40FF40) while the stroke of the polygon is set on **lines 8-9** to white (#FFFFFF) with a thickness of 2 pixels. The label is set in the `<TextSymbolizer>` on **lines 12-16**, with **line 14** determining what text to display, in this case the value of the "name" attribute. (Refer to the attribute table in the [Example polygons layer](#) section if necessary.) All other details about the label are set to the renderer default, which here is Times New Roman font, font color black, and font size of 10 pixels.

### Label halo

This example alters the look of the [Polygon with default label](#) by adding a white halo to the label.



*Label halo*

### Code

View and download the full "Label halo" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
5          <CssParameter name="fill">#40FF40</CssParameter>
6        </Fill>
7        <Stroke>
8          <CssParameter name="stroke">#FFFFFF</CssParameter>
9          <CssParameter name="stroke-width">2</CssParameter>
10        </Stroke>
11      </PolygonSymbolizer>
12      <TextSymbolizer>
13        <Label>
14          <ogc:PropertyName>name</ogc:PropertyName>
15        </Label>
16        <Halo>
17          <Radius>3</Radius>
18          <Fill>
19            <CssParameter name="fill">#FFFFFF</CssParameter>
```

```
20      </Fill>
21      </Halo>
22    </TextSymbolizer>
23    </Rule>
24  </FeatureTypeStyle>
```

### Details

This example is similar to the [Polygon with default label](#), with the addition of a halo around the labels on **lines 16-21**. A halo creates a color buffer around the label to improve label legibility. **Line 17** sets the radius of the halo, extending the halo 3 pixels around the edge of the label, and **line 19** sets the color of the halo to white (#FFFFFF). Since halos are most useful when set to a sharp contrast relative to the text color, this example uses a white halo around black text to ensure optimum readability.

### Polygon with styled label

This example improves the label style from the [Polygon with default label](#) example by centering the label on the polygon, specifying a different font name and size, and setting additional label placement optimizations.

### Note

The label placement optimizations discussed below (the `<VendorOption>` tags) are SLD extensions that are custom to GeoServer. They are not part of the SLD 1.0 specification.



*Polygon with styled label*

### Code

View and download the full "Polygon with styled label" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <PolygonSymbolizer>
4        <Fill>
5          <CssParameter name="fill">#40FF40</CssParameter>
6        </Fill>
7        <Stroke>
8          <CssParameter name="stroke">#FFFFFF</CssParameter>
9          <CssParameter name="stroke-width">2</CssParameter>
10        </Stroke>
11      </PolygonSymbolizer>
12      <TextSymbolizer>
13        <Label>
14          <ogc:PropertyName>name</ogc:PropertyName>
15        </Label>
16        <Font>
17          <CssParameter name="font-family">Arial</CssParameter>
18          <CssParameter name="font-size">11</CssParameter>
19          <CssParameter name="font-style">normal</CssParameter>
```

```
20      <CssParameter name="font-weight">bold</CssParameter>
21    </Font>
22    <LabelPlacement>
23      <PointPlacement>
24        <AnchorPoint>
25          <AnchorPointX>0.5</AnchorPointX>
26          <AnchorPointY>0.5</AnchorPointY>
27        </AnchorPoint>
28      </PointPlacement>
29    </LabelPlacement>
30    <Fill>
31      <CssParameter name="fill">#000000</CssParameter>
32    </Fill>
33    <VendorOption name="autoWrap">60</VendorOption>
34    <VendorOption name="maxDisplacement">150</VendorOption>
35  </TextSymbolizer>
36 </Rule>
37 </FeatureTypeStyle>
```

### Details

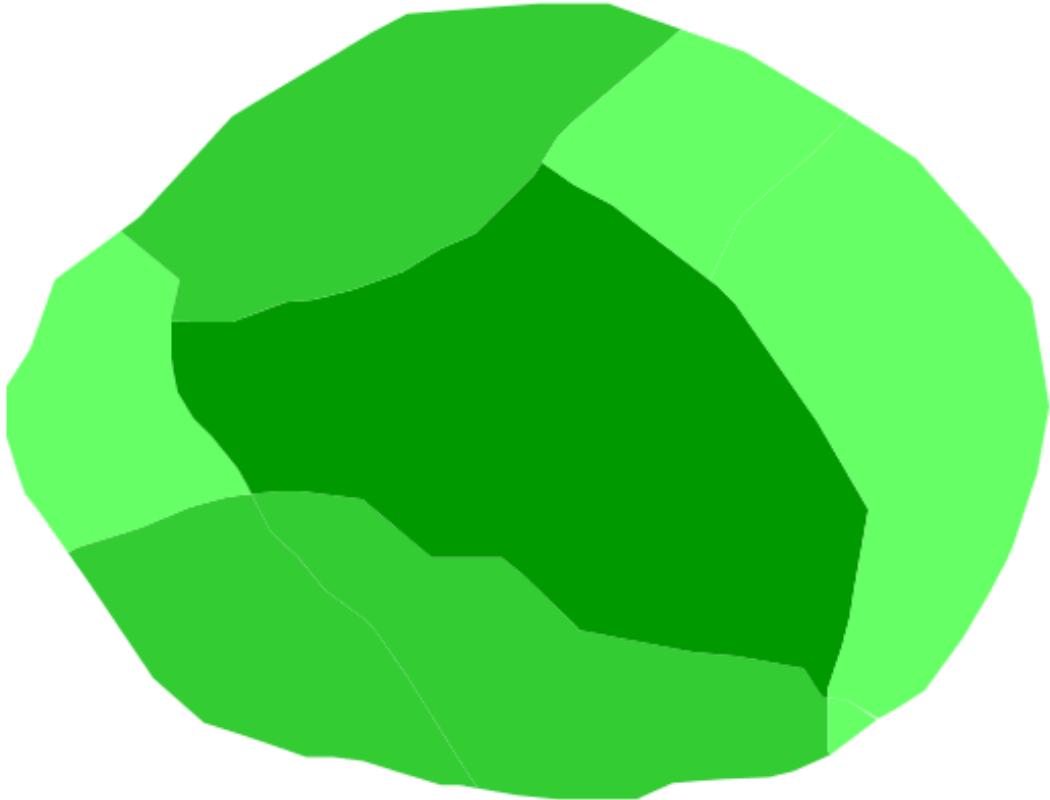
This example is similar to the [Polygon with default label](#) example, with additional styling options within the `<TextSymbolizer>` on lines **12-35**. **Lines 16-21** set the font styling. **Line 17** sets the font family to be Arial, **line 18** sets the font size to 11 pixels, **line 19** sets the font style to "normal" (as opposed to "italic" or "oblique"), and **line 20** sets the font weight to "bold" (as opposed to "normal").

The `<LabelPlacement>` tag on **lines 22-29** affects where the label is placed relative to the centroid of the polygon. **Line 21** centers the label by positioning it 50% (or 0.5) of the way horizontally along the centroid of the polygon. **Line 22** centers the label vertically in exactly the same way.

Finally, there are two added touches for label placement optimization: **line 33** ensures that long labels are split across multiple lines by setting line wrapping on the labels to 60 pixels, and **line 34** allows the label to be displaced by up to 150 pixels. This ensures that labels are compacted and less likely to spill over polygon boundaries. Notice little Massive County in the corner, whose label is now displayed."

### Attribute-based polygon

This example styles the polygons differently based on the "pop" (Population) attribute.



*Attribute-based polygon*

#### **Code**

View and download the full "Attribute-based polygon" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <Name>SmallPop</Name>
4      <Title>Less Than 200,000</Title>
5      <ogc:Filter>
6        <ogc:PropertyIsLessThan>
7          <ogc:PropertyName>pop</ogc:PropertyName>
8          <ogc:Literal>200000</ogc:Literal>
9        </ogc:PropertyIsLessThan>
10      </ogc:Filter>
11      <PolygonSymbolizer>
12        <Fill>
13          <CssParameter name="fill">#66FF66</CssParameter>
14        </Fill>
15      </PolygonSymbolizer>
16    </Rule>
17    <Rule>
18      <Name>MediumPop</Name>
19      <Title>200,000 to 500,000</Title>
```

```
20      <ogc:Filter>
21          <ogc:And>
22              <ogc:PropertyIsGreaterThanOrEqualTo>
23                  <ogc:PropertyName>pop</ogc:PropertyName>
24                  <ogc:Literal>200000</ogc:Literal>
25              </ogc:PropertyIsGreaterThanOrEqualTo>
26              <ogc:PropertyIsLessThan>
27                  <ogc:PropertyName>pop</ogc:PropertyName>
28                  <ogc:Literal>500000</ogc:Literal>
29              </ogc:PropertyIsLessThan>
30          </ogc:And>
31      </ogc:Filter>
32      <PolygonSymbolizer>
33          <Fill>
34              <CssParameter name="fill">#33CC33</CssParameter>
35          </Fill>
36      </PolygonSymbolizer>
37  </Rule>
38  <Rule>
39      <Name>LargePop</Name>
40      <Title>Greater Than 500,000</Title>
41      <ogc:Filter>
42          <ogc:PropertyIsGreaterThanOrEqualTo>
43              <ogc:PropertyName>pop</ogc:PropertyName>
44              <ogc:Literal>500000</ogc:Literal>
45          </ogc:PropertyIsGreaterThanOrEqualTo>
46      </ogc:Filter>
47      <PolygonSymbolizer>
48          <Fill>
49              <CssParameter name="fill">#009900</CssParameter>
50          </Fill>
51      </PolygonSymbolizer>
52  </Rule>
53 </FeatureTypeStyle>
```

### Details

#### Note

Refer to the [Example polygons layer](#) to see the attributes for the layer. This example has eschewed labels in order to simplify the style, but you can refer to the example [Polygon with styled label](#) to see which attributes correspond to which polygons.

Each polygon in our fictional country has a population that is represented by the population ("pop") attribute. This style contains three rules that alter the fill based on the value of "pop" attribute, with smaller values yielding a lighter color and larger values yielding a darker color.

The three rules are designed as follows:

Rule order	Rule name	Population ("pop")	Color
------------	-----------	--------------------	-------

Une réponse efficace à la directive INSPIRE

1	SmallPop	Less than 200,000	#66FF66
2	MediumPop	200,000 to 500,000	#33CC33
3	LargePop	Greater than 500,000	#009900

The order of the rules does not matter in this case, since each shape is only rendered by a single rule.

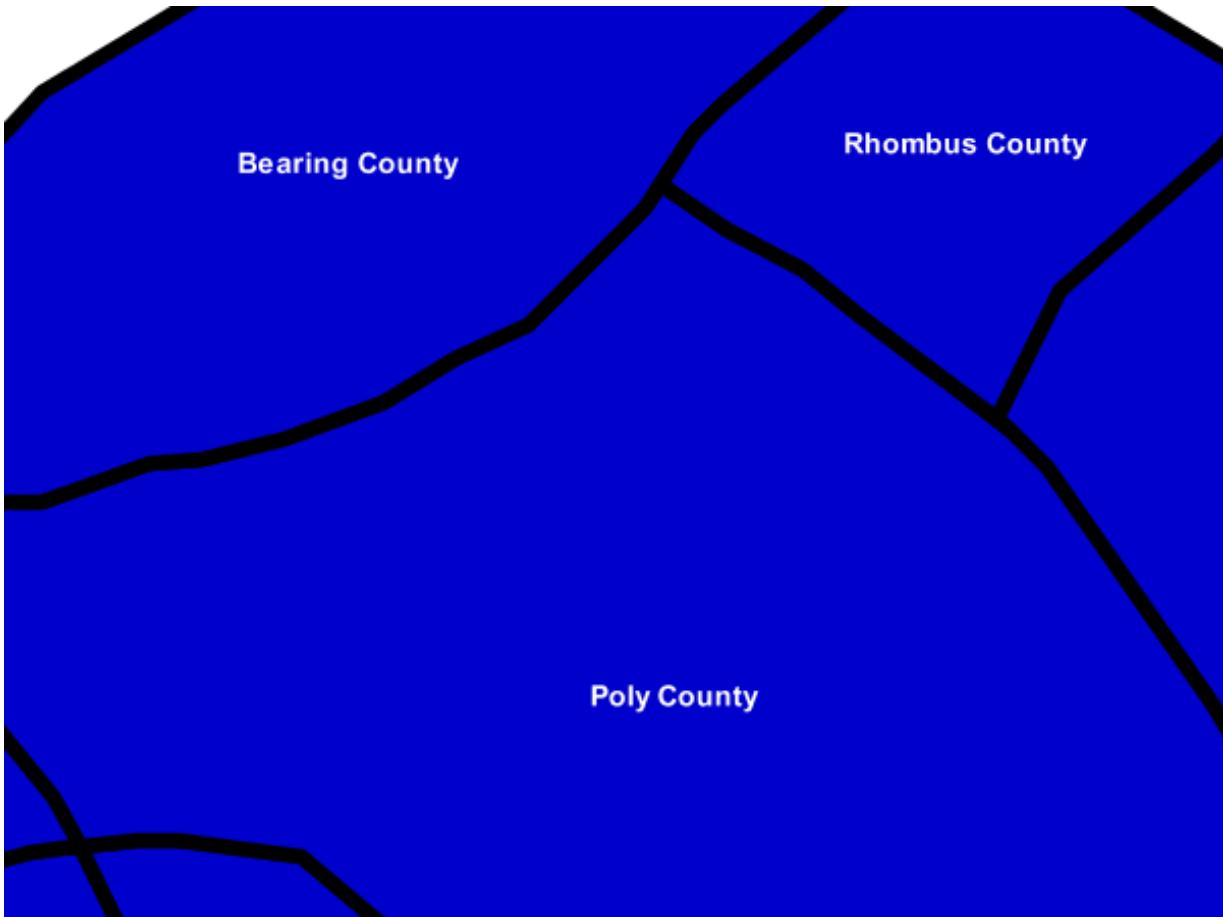
The first rule, on **lines 2-16**, specifies the styling of polygons whose population attribute is less than 200,000. **Lines 5-10** set this filter, with **lines 6-9** setting the "less than" filter, **line 7** denoting the attribute ("pop"), and **line 8** the value of 200,000. The color of the polygon fill is set to a light green (#66FF66) on **line 13**.

The second rule, on **lines 17-37**, is similar, specifying a style for polygons whose population attribute is greater than or equal to 200,000 but less than 500,000. The filter is set on **lines 20-31**. This filter is longer than in the first rule because two criteria need to be specified instead of one: a "greater than or equal to" and a "less than" filter. Notice the **And** on **line 21** and **line 30**. This mandates that both filters need to be true for the rule to be applicable. The color of the polygon fill is set to a medium green on (#33CC33) on **line 34**.

The third rule, on **lines 38-52**, specifies a style for polygons whose population attribute is greater than or equal to 500,000. The filter is set on **lines 41-46**. The color of the polygon fill is the only other difference in this rule, which is set to a dark green (#009900) on **line 49**.

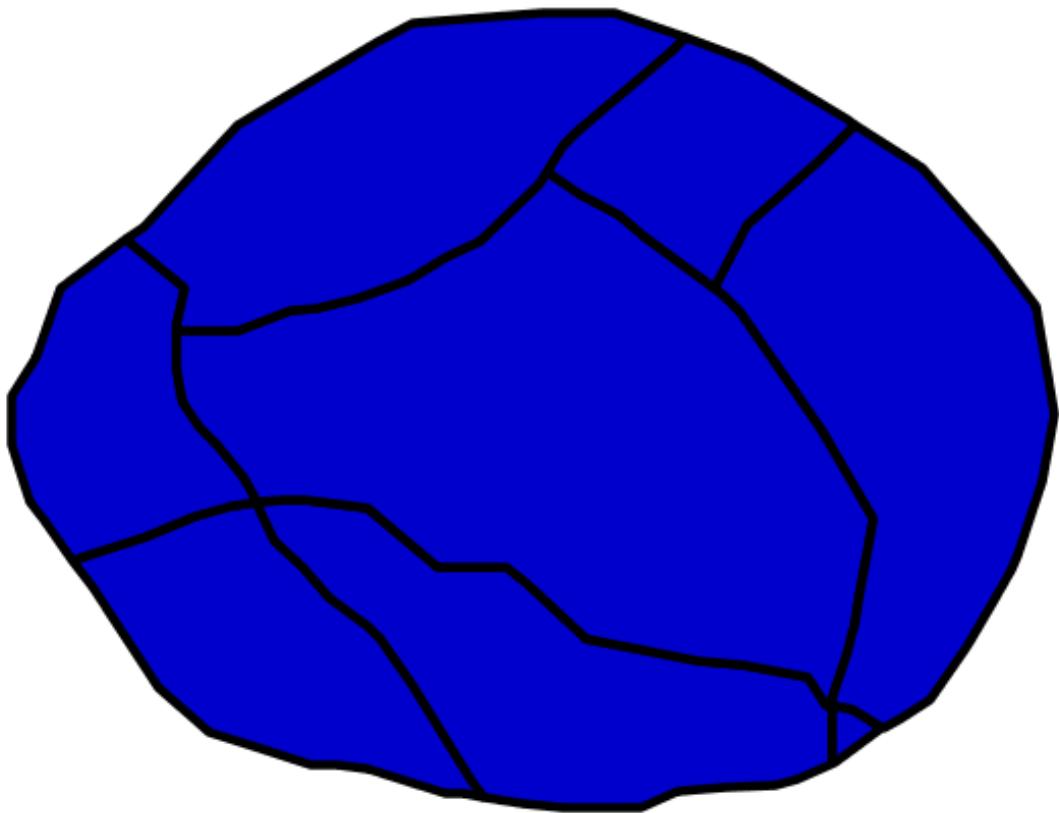
### **Zoom-based polygon**

This example alters the style of the polygon at different zoom levels.

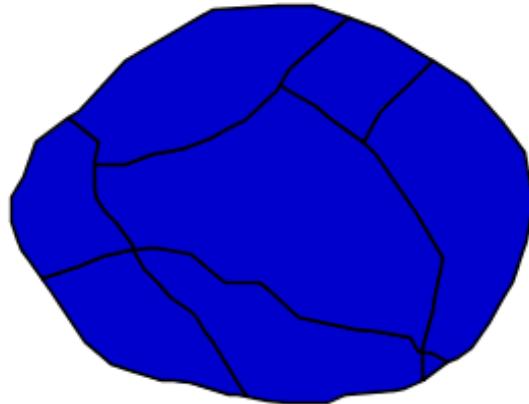


Une réponse efficace à la directive INSPIRE

*Zoom-based polygon: Zoomed in*



*Zoom-based polygon: Partially zoomed*



*Zoom-based polygon: Zoomed out*

### Code

View and download the full "Zoom-based polygon" SLD

```
1  <FeatureTypeStyle>
2    <Rule>
3      <Name>Large</Name>
4      <MaxScaleDenominator>100000000</MaxScaleDenominator>
5      <PolygonSymbolizer>
6        <Fill>
7          <CssParameter name="fill">#0000CC</CssParameter>
8        </Fill>
9        <Stroke>
10       <CssParameter name="stroke">#000000</CssParameter>
11       <CssParameter name="stroke-width">7</CssParameter>
12     </Stroke>
13   </PolygonSymbolizer>
14   <TextSymbolizer>
15     <Label>
16       <ogc:PropertyName>name</ogc:PropertyName>
17     </Label>
18     <Font>
19       <CssParameter name="font-family">Arial</CssParameter>
```

```

20      <CssParameter name="font-size">14</CssParameter>
21      <CssParameter name="font-style">normal</CssParameter>
22      <CssParameter name="font-weight">bold</CssParameter>
23  </Font>
24  <LabelPlacement>
25    <PointPlacement>
26      <AnchorPoint>
27        <AnchorPointX>0.5</AnchorPointX>
28        <AnchorPointY>0.5</AnchorPointY>
29      </AnchorPoint>
30    </PointPlacement>
31  </LabelPlacement>
32  <Fill>
33    <CssParameter name="fill">#FFFFFF</CssParameter>
34  </Fill>
35  </TextSymbolizer>
36 </Rule>
37 <Rule>
38   <Name>Medium</Name>
39   <MinScaleDenominator>100000000</MinScaleDenominator>
40   <MaxScaleDenominator>200000000</MaxScaleDenominator>
41   <PolygonSymbolizer>
42     <Fill>
43       <CssParameter name="fill">#0000CC</CssParameter>
44     </Fill>
45     <Stroke>
46       <CssParameter name="stroke">#000000</CssParameter>
47       <CssParameter name="stroke-width">4</CssParameter>
48     </Stroke>
49   </PolygonSymbolizer>
50 </Rule>
51 <Rule>
52   <Name>Small</Name>
53   <MinScaleDenominator>200000000</MinScaleDenominator>
54   <PolygonSymbolizer>
55     <Fill>
56       <CssParameter name="fill">#0000CC</CssParameter>
57     </Fill>
58     <Stroke>
59       <CssParameter name="stroke">#000000</CssParameter>
60       <CssParameter name="stroke-width">1</CssParameter>
61     </Stroke>
62   </PolygonSymbolizer>
63 </Rule>
64 </FeatureTypeStyle>

```

### **Details**

It is often desirable to make shapes larger at higher zoom levels when creating a natural-looking map. This example varies the thickness of the lines according to the zoom level. Polygons already do this by nature of being two dimensional, but another way to adjust styling of polygons based on zoom level is to adjust the thickness of the stroke (to be larger as the map is zoomed in) or to limit labels to only certain zoom levels. This ensures that the size and quantity of strokes and labels remains legible and doesn't overshadow the

polygons themselves.

Zoom levels (or more accurately, scale denominators) refer to the scale of the map. A scale denominator of 10,000 means the map has a scale of 1:10,000 in the units of the map projection.

### Note

Determining the appropriate scale denominators (zoom levels) to use is beyond the scope of this example.

This style contains three rules, defined as follows:

Rule order	Rule name	Scale denominator	Stroke width	Label display?
1	Large	1:100,000,000 or less	7	Yes
2	Medium	1:100,000,000 to 1:200,000,000	4	No
3	Small	Greater than 1:200,000,000	2	No

The first rule, on [lines 2-36](#), is for the smallest scale denominator, corresponding to when the view is "zoomed in". The scale rule is set on [line 40](#) such that the rule will apply only where the scale denominator is 100,000,000 or less. [Line 7](#) defines the fill as blue (#0000CC). Note that the fill is kept constant across all rules regardless of the scale denominator. As in the [Polygon with default label](#) or [Polygon with styled label](#) examples, the rule also contains a <TextSymbolizer> at [lines 14-35](#) for drawing a text label on top of the polygon. [Lines 19-22](#) set the font information to be Arial, 14 pixels, and bold with no italics. The label is centered both horizontally and vertically along the centroid of the polygon on by setting <AnchorPointX> and <AnchorPointY> to both be 0.5 (or 50%) on [lines 27-28](#). Finally, the color of the font is set to white (#FFFFFF) in [line 33](#).

The second rule, on [lines 37-50](#), is for the intermediate scale denominators, corresponding to when the view is "partially zoomed". The scale rules on [lines 39-40](#) set the rule such that it will apply to any map with a scale denominator between 100,000,000 and 200,000,000. (The <MinScaleDenominator> is inclusive and the <MaxScaleDenominator> is exclusive, so a zoom level of exactly 200,000,000 would *not* apply here.) Aside from the scale, there are two differences between this rule and the first: the width of the stroke is set to 4 pixels on [line 47](#) and a <TextSymbolizer> is not present so that no labels will be displayed.

The third rule, on [lines 51-63](#), is for the largest scale denominator, corresponding to when the map is "zoomed out". The scale rule is set on [line 53](#) such that the rule will apply to any map with a scale denominator of 200,000,000 or greater. Again, the only differences between this rule and the others are the width of the lines, which is set to 1 pixel on [line 60](#), and the absence of a <TextSymbolizer> so that no labels will be displayed.

The resulting style produces a polygon stroke that gets larger as one zooms in and labels that only display when zoomed in to a sufficient level.

### Rasters

Les rasters sont des données géographiques disposées dans une grille. Elles sont similaires à des images comme les fichiers PNG, mais au lieu que chaque point contienne des informations visuelles, il contient des informations géographiques sous forme numérique. Les rasters peuvent être vus comme une matrice géoréférencée de valeurs numériques.

## Une réponse efficace à la directive INSPIRE

Un exemple de raster est une couche modèle numérique d'élévation (DEM), qui possède des données d'élévation codées sous forme d'un à chaque point géoréférencé.

### Avis

Pour rester concis, les exemples de code présentés sur cette page ne sont **pas le code SLD complet** car ils omettent les informations SLD de début et de fin. Utilisez les liens pour télécharger les SLD complet de chaque exemple.

### Exemple de raster

La couche raster utilisée dans les exemples ci-dessous contient les données d'élévation d'un monde fictif. Les données sont stockées en projection EPSG:4326 (longitude/latitude) et des extrêmes de 70 à 256. Si elles sont restituées en niveaux de gris, où les valeurs minimum sont colorées en noir et les valeurs maximum en blanc, le raster ressemblerait à ceci:



*Fichier raster restitué en niveaux de gris*

Téléchargez le raster et le shapefile

### Gradient bicolore

Cet exemple montre un style bicolore avec du vert à basse altitude et du brun à haute altitude.



*Gradient bicolore*

### **Code**

Consultez et téléchargez le SLD complet "Gradient bicolore"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ColorMap>
5          <ColorMapEntry color="#008000" quantity="70" />
6          <ColorMapEntry color="#663333" quantity="256" />
7        </ColorMap>
8      </RasterSymbolizer>
9    </Rule>
10   </FeatureTypeStyle>
```

### **Détails**

Dans cet exemple, il y a un `<Rule>` dans un `<FeatureTypeStyle>` ce qui est la situation la plus simple possible. Tous les exemples suivants partageront cette caractéristique. La symbolisation des rasters est effectuée avec la balise `<RasterSymbolizer>` (**lignes 3-8**).

Cet exemple crée un gradient régulier entre deux couleurs correspondant à deux valeurs d'élévation. Le gradient est créé avec `<ColorMap>` en **lignes 4-7**. Chaque entrée dans `<ColorMap>` représente une entrée, ou "ancre", dans le gradient. La **ligne 5** règle la valeur basse égale à 70 via le paramètre `quantity`, valeur représentée en vert foncé (#008000). La **ligne 6** règle la valeur haute de 256, à nouveau via le paramètre `quantity`, et la valeur est représentée en brun foncé (#663333). Toutes les valeurs entre ces deux quantités sont interpolées linéairement: une valeur de 163 (le point médian entre 70 et 256) sera colorée avec une teinte médiane entre les deux couleurs (dans ce cas, environ #335717, un brun-vert).

### **Gradient transparent**

## Une réponse efficace à la directive INSPIRE

Cet exemple crée le même gradient bicolore que [Gradient bicolore](#) de l'exemple précédent, mais rend la couche entière presque transparente en réglant une opacité à 30%.



*Gradient transparent*

### Code

Consultez et téléchargez le SLD complet "Transparent gradient"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <Opacity>0.3</Opacity>
5        <ColorMap>
6          <ColorMapEntry color="#008000" quantity="70" />
7          <ColorMapEntry color="#663333" quantity="256" />
8        </ColorMap>
9      </RasterSymbolizer>
10     </Rule>
11   </FeatureTypeStyle>
```

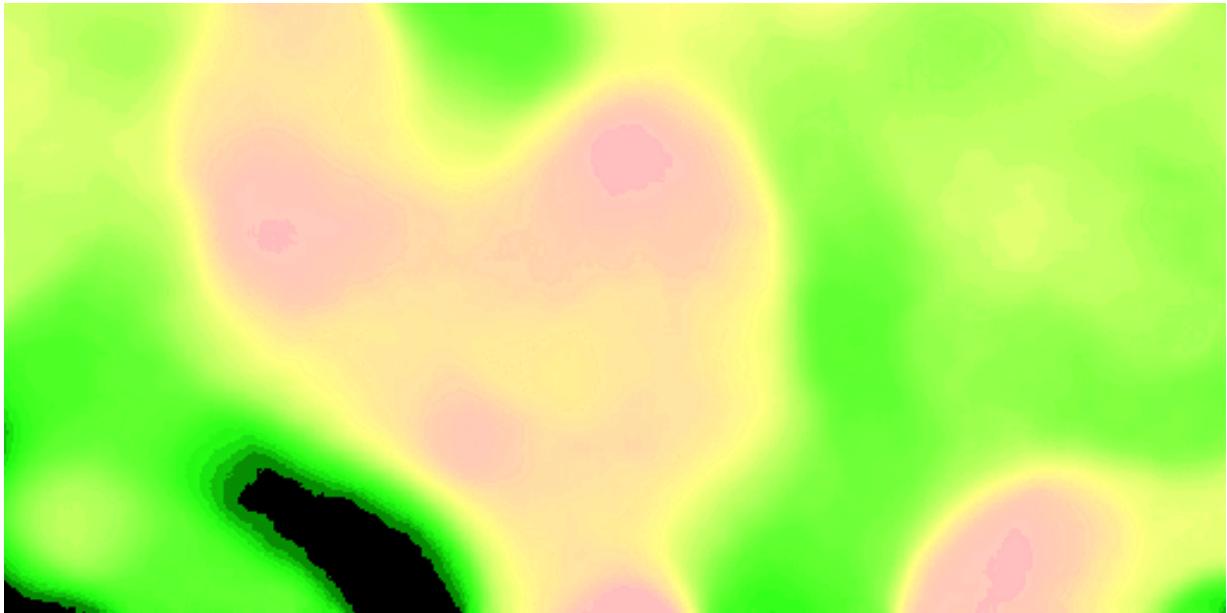
### Détails

Cet exemple est similaire à l'exemple [Gradient bicolore](#) mis à part l'ajout de la **ligne 4**, laquelle règle l'opacité de la couche à 0.3 (ou opaque à 30%). Une valeur d'opacité de 1 signifie que la forme est dessinée 100% opaque, alors qu'une valeur de 0 signifie qu'elle est affichée entièrement transparente. Une valeur de 0.3 signifie que le raster prend partiellement la couleur et le style de ce qui est dessiné en-dessous. Comme le fond est blanc dans cet exemple, les couleurs générées par `<ColorMap>` paraissent plus claires, mais si le raster est superposé à un fond sombre la couleur résultante sera plus sombre.

### Luminosité et contraste

Cet exemple normalise les couleurs de sortie, puis augmentent la luminosité d'un facteur 2.

Une réponse efficace à la directive INSPIRE



Luminosité et contraste

### Code

Consultez et téléchargez le SLD complet "Luminosité et contraste"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ContrastEnhancement>
5          <Normalize />
6          <GammaValue>0.5</GammaValue>
7        </ContrastEnhancement>
8        <ColorMap>
9          <ColorMapEntry color="#008000" quantity="70" />
10         <ColorMapEntry color="#663333" quantity="256" />
11       </ColorMap>
12     </RasterSymbolizer>
13   </Rule>
14 </FeatureTypeStyle>
```

### Détails

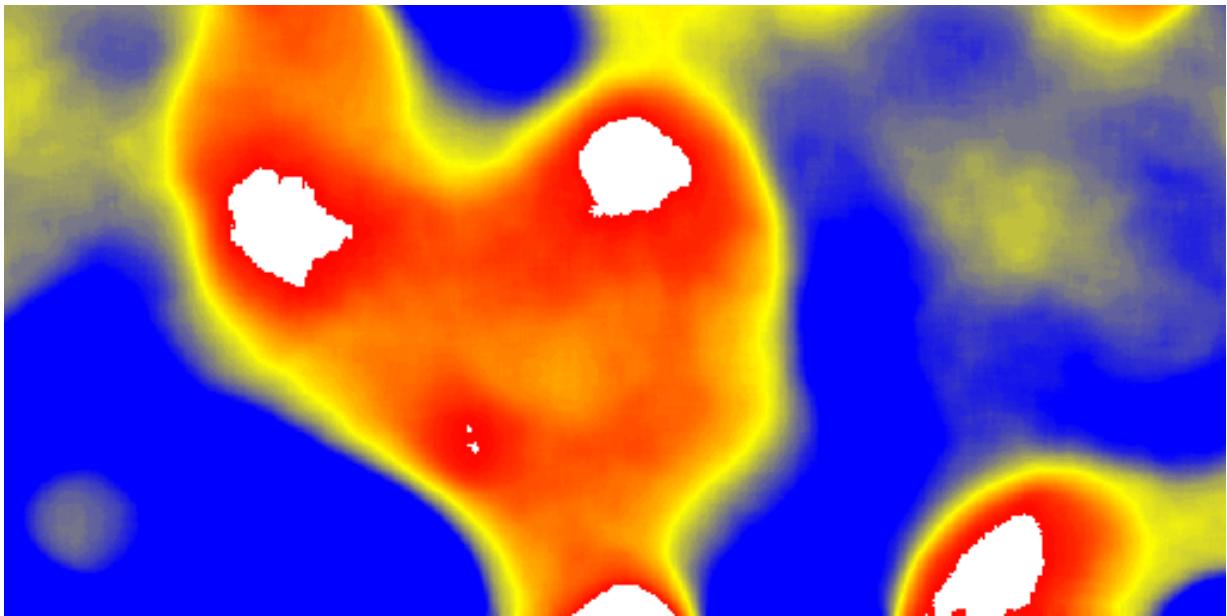
Cet exemple est similaire à [Gradient bicolore](#), mis à part l'ajout de la balise `<ContrastEnhancement>` en **lignes 4-7**. La **ligne 5** normalise la sortie en augmentant le contraste à son extension maximum. La **ligne 6** ajuste la luminosité d'un facteur 0.5. Comme les valeurs inférieures à 1 rendent le résultat plus clair, une valeur de 0.5 rend le résultat deux fois plus clair.

Comme dans les exemples précédents, les **lignes 8-11** déterminent la `<ColorMap>`, avec la **ligne 9** paramétrant la limite inférieure (70) à une couleur vert foncé (#008000) et la **ligne 10** paramétrant la limite supérieure (256) à une couleur marron foncé (#663333).

### Gradient à trois couleurs

## Une réponse efficace à la directive INSPIRE

Cet exemple crée un gradient à trois couleurs prises dans les couleurs primaires. De plus, le gradient ne prend pas en compte l'étendue complète des valeurs des données, ce qui aboutit à ne pas prendre en compte certaines données.



*Gradient à trois couleurs*

### Code

Consultez et téléchargez le SLD complet "Three-color gradient"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ColorMap>
5          <ColorMapEntry color="#0000FF" quantity="150" />
6          <ColorMapEntry color="#FFFF00" quantity="200" />
7          <ColorMapEntry color="#FF0000" quantity="250" />
8        </ColorMap>
9      </RasterSymbolizer>
10    </Rule>
11  </FeatureTypeStyle>
```

### Détails

Cet exemple crée un gradient à trois couleurs basé sur une `<ColorMap>` avec trois entrées en lignes 4-8: la ligne 5 paramétrant la limite inférieure (150) à bleu (#0000FF), la ligne 6 paramétrant une valeur intermédiaire (200) à jaune (#FFFF00), et la ligne 7 paramétrant la limite supérieure (250) à rouge (#FF0000).

Comme nos valeurs de données vont de 70 à 256, certains points ne sont pas pris en compte par ce style. Les valeurs inférieures à la limite basse de la carte de couleurs (l'intervalle 70 à 149) sont rendues avec la même couleurs que la limite basse, à savoir bleu. D'autre part, les valeurs supérieures à la limite haute de la carte de couleur (l'intervalle 251 à 256) ne sont pas affichées du tout.

### Canal alpha

## Une réponse efficace à la directive INSPIRE

Cet exemple crée un effet de "canal alpha" de sorte que les valeurs supérieures sont graduellement transparentes.



*Canal alpha*

### Code

Consultez et téléchargez le SLD complet "Alpha channel"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ColorMap>
5          <ColorMapEntry color="#008000" quantity="70" />
6          <ColorMapEntry color="#008000" quantity="256" opacity="0"/>
7        </ColorMap>
8      </RasterSymbolizer>
9    </Rule>
10   </FeatureTypeStyle>
```

### Détails

Un canal alpha est une autre façon d'aboutir à une transparence variable. Tout comme les gradients lient les valeurs aux couleurs, chaque entrée de `<ColorMap>` peut avoir des valeurs d'opacité (la valeur par défaut étant 1.0 soit complètement opaque).

Dans cet exemple, il y a une `<ColorMap>` avec deux entrées: la **ligne 5** paramètre la limite inférieure de 70 à vert foncé (#008000), et la **ligne 6** spécifie la limite supérieure de 256 à vert foncé également, mais avec une valeur d'opacité de 0. Cela signifie que les valeurs à 256 seront affichées avec une opacité de 0% (entièremment transparent). Tout comme le gradient de couleurs, l'opacité est aussi interpolée linéairement, de sorte qu'une valeur de 163 (le point médian entre 70 et 256) sera affichée avec une opacité de 50%.

### Couleurs discrètes

Cet exemple présente un gradient qui n'est pas interpolé linéairement, mais qui à la place fait correspondre précisément des valeurs à trois couleurs.

### Note

Cet exemple met en exergue une extension SLD propre à GeoServer. Les couleurs discrètes ne font pas partie de la spécification SLD.



*Couleurs discrètes*

### Code

Consultez et téléchargez le SLD complet "Couleurs discrètes"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ColorMap type="intervals">
5          <ColorMapEntry color="#008000" quantity="150" />
6          <ColorMapEntry color="#663333" quantity="256" />
7        </ColorMap>
8      </RasterSymbolizer>
9    </Rule>
10   </FeatureTypeStyle>
```

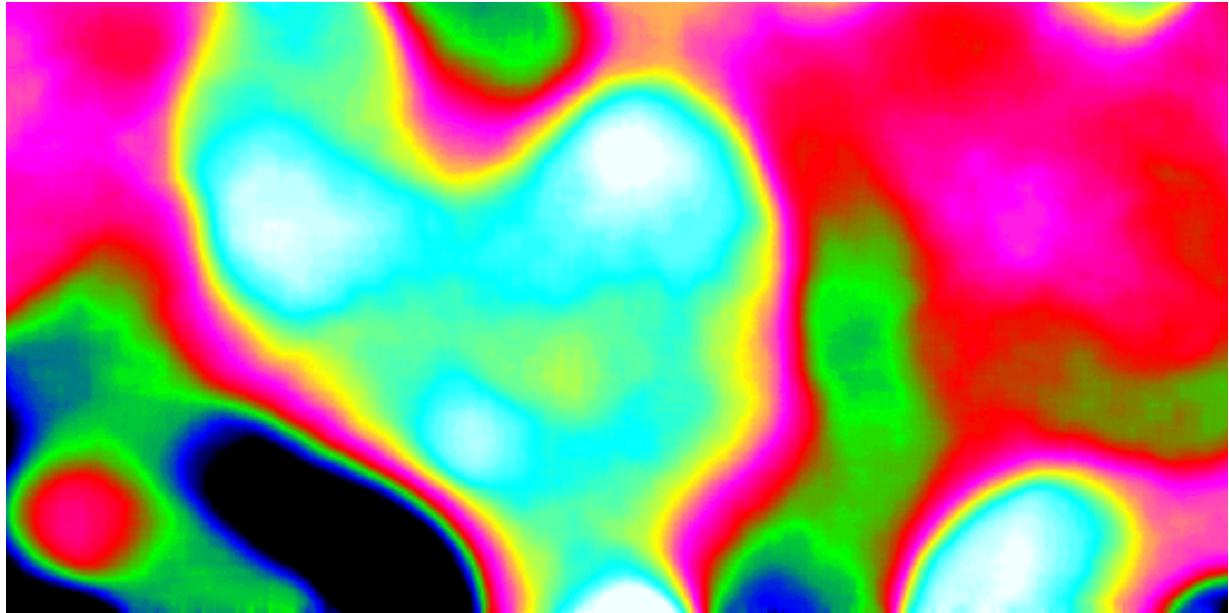
### Détails

Parfois, des bandes de couleur distribuées par paliers discrets sont plus appropriées que les gradients de couleur. Le paramètre `type="intervals"` ajouté à `<ColorMap>` en **ligne 4** indique l'emploi de couleurs discrètes à la place d'un gradient. Les valeurs dans chaque entrée correspondent à la limite supérieure pour la bande de couleur, de sorte que les couleurs sont reliées aux valeurs inférieures à la valeur d'une entrée, et supérieures ou égales à l'entrée suivante. Par exemple, la **ligne 5** colorie toutes les valeurs inférieures à 150 en vert foncé (#008000) et la **ligne 6** colorie toutes les valeurs inférieures à 256 et supérieures ou égales à 150 en brun foncé (#663333).

Une réponse efficace à la directive INSPIRE

### **Gradient multicolore**

Cet exemple présente un gradient à huit couleurs.



*Gradient multicolore*

### **Code**

Consultez et téléchargez le SLD complet "Gradient multicolore"

```
1  <FeatureTypeStyle>
2    <Rule>
3      <RasterSymbolizer>
4        <ColorMap>
5          <ColorMapEntry color="#000000" quantity="95" />
6          <ColorMapEntry color="#0000FF" quantity="110" />
7          <ColorMapEntry color="#00FF00" quantity="135" />
8          <ColorMapEntry color="#FF0000" quantity="160" />
9          <ColorMapEntry color="#FF00FF" quantity="185" />
10         <ColorMapEntry color="#FFFF00" quantity="210" />
11         <ColorMapEntry color="#00FFFFFF" quantity="235" />
12         <ColorMapEntry color="#FFFFFF" quantity="256" />
13       </ColorMap>
14     </RasterSymbolizer>
15   </Rule>
16 </FeatureTypeStyle>
```

### **Détails**

Il n'y a pas de limite au nombre d'entrées contenues dans `<ColorMap>` (**lignes 4-13**). Cet exemple a huit entrées:

<b>Entry number</b>	<b>Value</b>	<b>Color</b>	<b>RGB code</b>
1	95	Black	#000000
2	110	Blue	#0000FF

## Une réponse efficace à la directive INSPIRE

3	135	Green	#00FF00
4	160	Red	#FF0000
5	185	Purple	#FF00FF
6	210	Yellow	#FFFF00
7	235	Cyan	#00FFFF
8	256	White	#FFFFFF

### **SLD Reference**

A symbolizer specifies how data should be visualized. There are 5 types of symbolizers: PointSymbolizer, which is used to portray **point** data; LineSymbolizer, which is used to portray **line** data; PolygonSymbolizer, which is used to portray **polygon** data; RasterSymbolizer, which is used to portray **raster** data; and TextSymbolizer, which is used to portray **text labels**.

### **Avis**

Intro for filters and scale.

### **PointSymbolizer**

The PointSymbolizer styles **points**. Points are elements that contain only position information.

### **Syntax**

The outermost element is the <Graphic> tag. This determines the type of visualization. There are five possible tags to include inside the <Graphic> tag:

Tag	Required?	Description
<ExternalGraphic>	No (when using <Mark>)	Specifies an image file to use as the symbolizer.
<Mark>	No (when using <ExternalGraphic>)	Specifies a common shape to use as the symbolizer.
<Opacity>	No	Determines the opacity (transparency) of symbolizers. Values range from 0 (completely transparent) to 1 (completely opaque). Default is 1.
<Size>	Yes	Determines the size of the symbolizer in pixels. When used with an image file, this will specify the height of the image, with the width scaled accordingly.
<Rotation>	No	Determines the rotation of the graphic in degrees. The rotation increases in the clockwise direction. Negative values indicate counter-clockwise rotation. Default is 0.

Within the <ExternalGraphic> tag, there are also additional tags:

## Une réponse efficace à la directive INSPIRE

Tag	Required?	Description
<OnlineResource>	Yes	The location of the image file. Can be either a URL or a local path relative to the SLD.
<Format>	Yes	The MIME type of the image format. Most standard web image formats are supported.

Within the <Mark> tag, there are also additional tags:

Tag	Required?	Description
<WellKnownName>	Yes	The name of the common shape. Options are circle, square, triangle, star, cross, or x. Default is square.
<Fill>	No (when using <Stroke>)	Specifies how the symbolizer should be filled. Options are a <CssParameter name="fill"> specifying a color in the form #RRGGBB, or <GraphicFill> for a repeated graphic.
<Stroke>	No (when using <Fill>)	Specifies how the symbolizer should be drawn on its border. Options are a <CssParameter name="fill"> specifying a color in the form #RRGGBB or <GraphicStroke> for a repeated graphic.

### Example

Consider the following symbolizer taken from the Simple Point example in the *Points* section in the *Livre de recettes SLD*.

```

1   <PointSymbolizer>
2     <Graphic>
3       <Mark>
4         <WellKnownName>circle</WellKnownName>
5         <Fill>
6           <CssParameter name="fill">#FF0000</CssParameter>
7         </Fill>
8       </Mark>
9       <Size>6</Size>
10      </Graphic>
11    </PointSymbolizer>

```

The symbolizer contains a <Graphic> tag, which is required. Inside this tag is the <Mark> tag and <Size> tag, which are the minimum required tags inside <Graphic> (when not using the <ExternalGraphic> tag). The <Mark> tag contains the <WellKnownName> tag and a <Fill> tag. No other tags are required. In summary, this example specifies the following:

1. Data will be rendered as points
  2. Points will be rendered as circles
  3. Circles will be rendered with a diameter of 6 pixels and filled with the color red
- Further examples can be found in the *Points* section of the *Livre de recettes SLD*.

### LineSymbolizer

The LineSymbolizer styles **lines**. Lines are one-dimensional geometry elements that contain position and length. Lines can be comprised of multiple line segments.

### Syntax

The outermost tag is the `<Stroke>` tag. This tag is required, and determines the visualization of the line. There are three possible tags that can be included inside the `<Stroke>` tag.

Tag	Required?	Description
<code>&lt;GraphicFill&gt;</code>	No	Renders the pixels of the line with a repeated pattern.
<code>&lt;GraphicStroke&gt;</code>	No	Renders the line with a repeated linear graphic.
<code>&lt;CssParameter&gt;</code>	No	Determines the stroke styling parameters.

When using the `<GraphicStroke>` and `<GraphicFill>` tags, it is required to insert the `<Graphic>` tag inside them. The syntax for this tag is identical to that mentioned in the [PointSymbolizer](#) section above.

Within the `<CssParameter>` tag, there are also additional parameters that go inside the actual tag:

Parameter	Required?	Description
<code>name="stroke"</code>	No	Specifies the solid color given to the line, in the form <code>#RRGGBB</code> . Default is black ( <code>#000000</code> ).
<code>name="stroke-width"</code>	No	Specifies the width of the line in pixels. Default is 1.
<code>name="stroke-opacity"</code>	No	Specifies the opacity (transparency) of the line. possible values are between 0 (completely transparent) and 1 (completely opaque). Default is 1.
<code>name="stroke-linejoin"</code>	No	Determines how lines are rendered at intersections of line segments. Possible values are <code>mitre</code> (sharp corner), <code>round</code> (rounded corner), and <code>bevel</code> (diagonal corner). Default is <code>mitre</code> .
<code>name="stroke-linecap"</code>	No	Determines how lines are rendered at ends of line segments. Possible values are <code>butt</code> (sharp square edge), <code>round</code> (rounded edge), and <code>square</code> (slightly elongated square edge). Default is <code>butt</code> .
<code>name="stroke-dasharray"</code>	No	Encodes a dash pattern as a series of numbers separated by spaces. Odd-indexed numbers (first, third, etc) determine the length in pixels to draw the line, and even-indexed numbers (second, fourth, etc) determine the length in pixels to blank out the line. Default is an unbroken line. <i>Starting from version 2.1</i> dash arrays can be combined with graphic strokes to generate complex line styles with alternating symbols or a mix of lines and symbols.
<code>name="stroke-dashoffset"</code>	No	Specifies the distance in pixels into the <code>dasharray</code> pattern at which to start drawing. Default is 0.

Une réponse efficace à la directive INSPIRE

### PolygonSymbolizer

The LineSymbolizer styles **polygons**. Lines are two-dimensional geometry elements. They can contain styling information about their border (stroke) and their fill.

#### Syntax

A <PolygonSymbolizer> can have two outermost tags:

Tag	Required?	Description
<Fill>	No (when using <Stroke>)	Determines the styling for the fill of the polygon.
<Stroke>	No (when using <Fill>)	Determines the styling for the stroke of the polygon.

The details for the <Stroke> tag are identical to that mentioned in the [LineSymbolizer](#) section above.

Within the <Fill> tag, there are additional tags:

Tag	Required?	Description
<GraphicFill>	No	Renders the fill of the polygon with a repeated pattern.
<CssParameter>	No	Determines the fill styling parameters.

When using the <GraphicFill> tag, it is required to insert the <Graphic> tag inside it. The syntax for this tag is identical to that mentioned in the [PointSymbolizer](#) section above.

Within the <CssParameter> tag, there are also additional parameters that go inside the actual tag:

Parameter	Required?	Description
name="fill"	No	Specifies the fill color for the polygon, in the form #RRGGBB. Default is grey (#808080).
name="fill-opacity"	No	Specifies the opacity (transparency) of the fill of the polygon. Possible values are between 0 (completely transparent) and 1 (completely opaque). Default is 1.

#### Example

Consider the following symbolizer taken from the Simple Point example in the [Polygons](#) section in the [Livre de recettes SLD](#).

```
1  <PolygonSymbolizer>
2    <Fill>
3      <CssParameter name="fill">#000080</CssParameter>
4    </Fill>
5  </PolygonSymbolizer>
```

This symbolizer contains only a <Fill> tag. Inside this tag is a <CssParameter> that specifies a fill color for the polygon to be #000080, or a muted blue.

Further examples can be found in the [Polygons](#) section of the [Livre de recettes SLD](#).

## Raster Symbolizer

### Introduction

GeoServer supports the ability to display raster data in addition to vector data.

Raster data is not merely a picture, rather it can be thought of as a grid of georeferenced information, much like a graphic is a grid of visual information (with combination of reds, greens, and blues). Unlike graphics, which only contain visual data, each point/pixel in a raster grid can have lots of different attributes, with possibly none of them having an inherently visual component.

With the above in mind, one needs to choose how to visualize the data, and this, like in all other cases, is done by using an SLD. The analogy to vector data is evident in the naming of the tags used. Vectors, consisting of points, line, and polygons, are styled by using the `<PointSymbolizer>`, `<LineSymbolizer>`, and `<PolygonSymbolizer>` tags. It is therefore not very surprising that raster data is styled with the tag `<RasterSymbolizer>`.

### Elements and Syntax

The following elements are available to be used as parameters inside `<RasterSymbolizer>`.

- `<Opacity>`
- `<ColorMap>`
- `<ChannelSelection>`
- `<ContrastEnhancement>`
- `<ShadedRelief>`
- `<OverlapBehavior>`
- `<ImageOutline>`

Notice that not all the above are actually implemented in the current version of the GeoServer.

### Opacity

This element sets the transparency level for the entire dataset. As is standard, the values range from zero (0) to one (1), with zero being totally transparent, and one being not transparent at all. The syntax for `<Opacity>` is very simple:

```
<Opacity>0.5</Opacity>
```

where, in this case, the raster would be displayed at 50% opacity.

### ColorMap

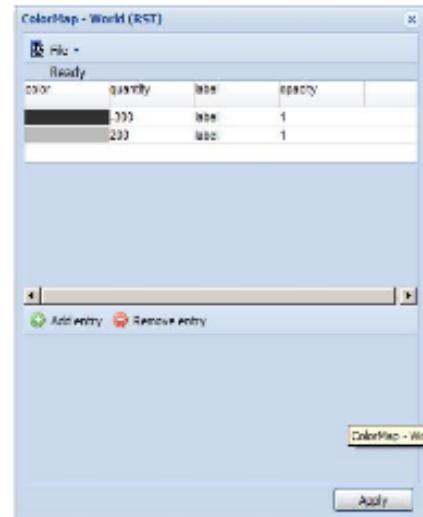
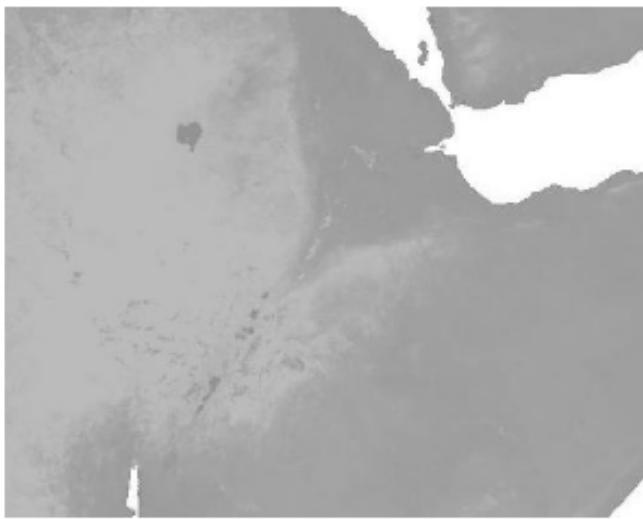
The `<ColorMap>` element sets rules for color gradients based on the quantity attribute. This quantity refers to the magnitude of the value of a data point. At its simplest, one could create two color map entries (the element called `<ColorMapEntry>`, one with a color for the "bottom" of the dataset, and another with a color for the "top" of the dataset. The colors in between will be automatically interpolated with the quantity values in between, making creating color gradients easy. One can also fine tune the color map by adding additional entries, which is handy if the dataset has more discrete values rather than a gradient. In that case, one could add an entry for each value to be set to a different color. In all cases, the color is denoted in standard hexadecimal RGB format (#RRGGBB). In addition to color and quantity, `ColorMapEntry` elements can also have opacity and label, the former which could

## Une réponse efficace à la directive INSPIRE

be used instead of the global value mentioned previously, and the latter which could be used for legends.

For example a simple ColorMap can be:

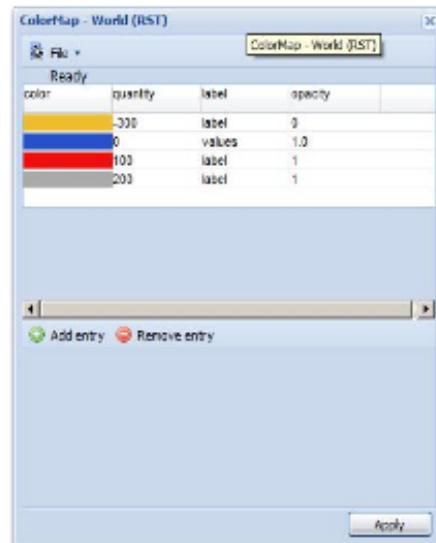
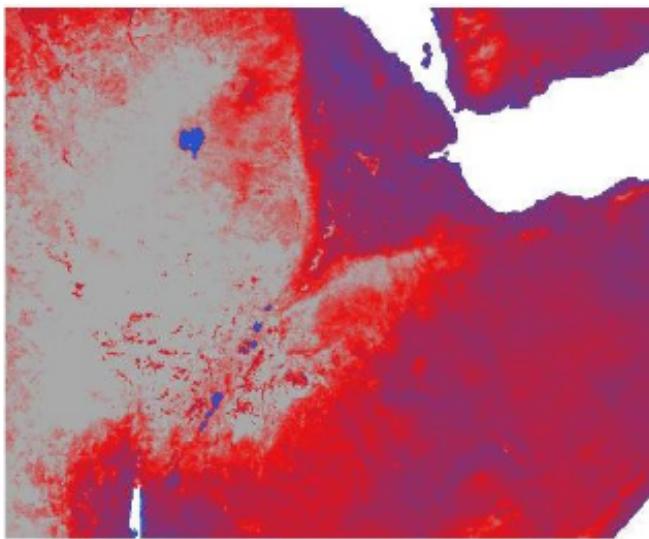
```
<ColorMap>
    <ColorMapEntry color="#323232" quantity="-300" label="label1" opacity="1"/>
    <ColorMapEntry color="#BBBBBB" quantity="200" label="label2" opacity="1"/>
</ColorMap>
```



This example would create a color gradient from #323232 color to #BBBBBB color using quantity values -300 to 200:

```
<ColorMap>
    <ColorMapEntry color="#FFCC32" quantity="-300" label="label1" opacity="0"/>
    <ColorMapEntry color="#3645CC" quantity="0" label="label2" opacity="1"/>
    <ColorMapEntry color="#CC3636" quantity="100" label="label3" opacity="1"/>
    <ColorMapEntry color="#BBBBBB" quantity="200" label="label4" opacity="1"/>
</ColorMap>
```

## Une réponse efficace à la directive INSPIRE

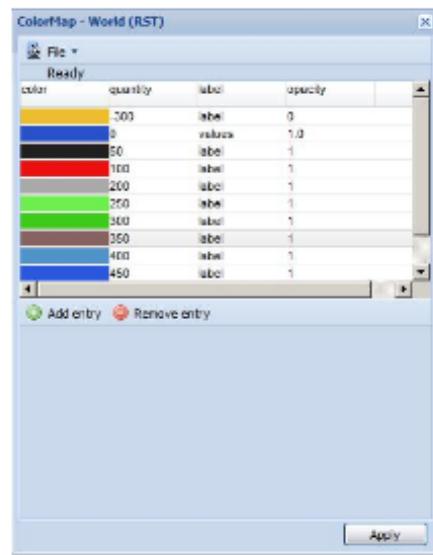
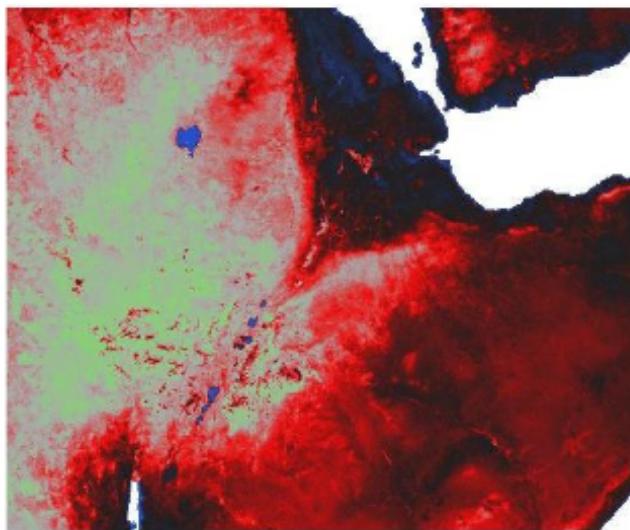


This example would create a color gradient from #FFCC32 color through #BBBBBB color running through #3645CC color and #CC3636 color. Here, though, #FFCC32 color would be transparent (simulating an alpha channel). Notice that default opacity, when not specified, is 1, which means opaque.

Two attributes can be created in ColorMap root node like 'type' and 'extended'.

The 'type' attribute specifies the kind of ColorMap to use. There are three different types of ColorMaps that can be specified through this attribute: ramp, intervals and values.

The 'ramp' is the default ColorMap type and the outcome is like the one presented at the beginning of this section (if into the ColorMap tag the attribute 'type' is not specified, the default value is 'ramp').

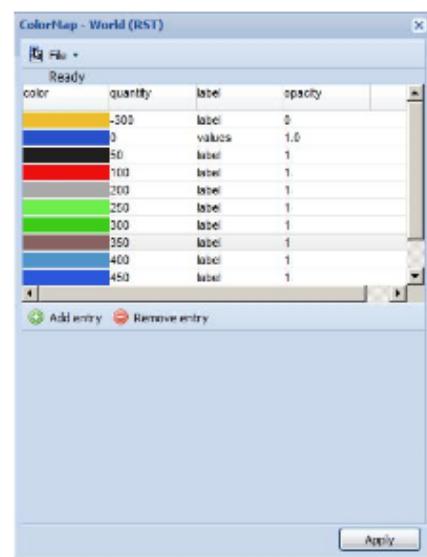


The 'values' means that only the specified entry quantities will be rendered, i.e. no color interpolation is applied between the entries.

The following example can clarify this aspect:

## Une réponse efficace à la directive INSPIRE

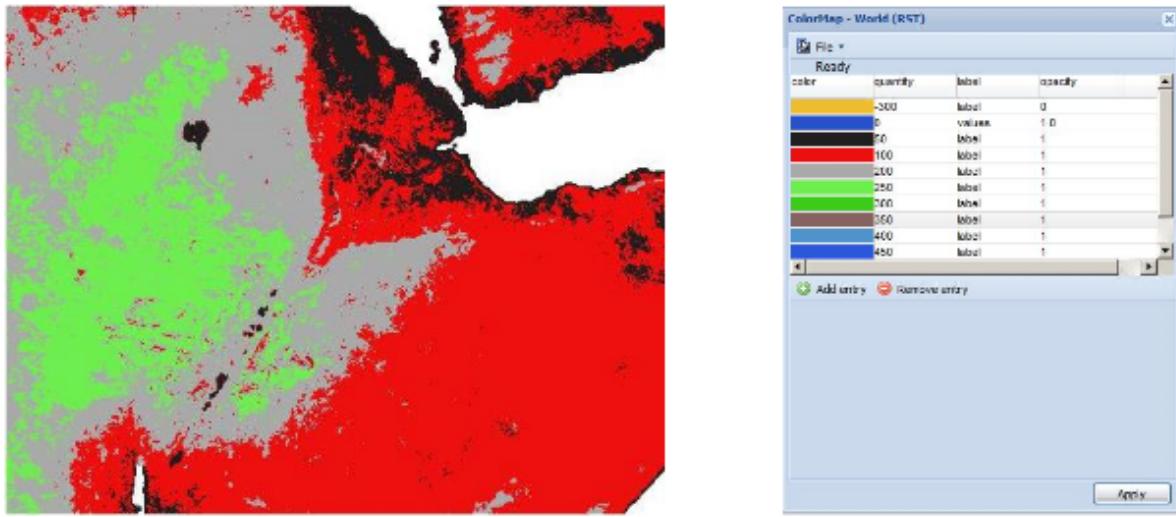
```
<ColorMap type="values">
    <ColorMapEntry color="#EEBE2F" quantity="-300" label="label" opacity="0"/>
    <ColorMapEntry color="#2851CC" quantity="0" label="values" opacity="1"/>
    <ColorMapEntry color="#211F1F" quantity="50" label="label" opacity="1"/>
    <ColorMapEntry color="#EE0F0F" quantity="100" label="label" opacity="1"/>
    <ColorMapEntry color="#AAAAAA" quantity="200" label="label" opacity="1"/>
    <ColorMapEntry color="#6FEE4F" quantity="250" label="label" opacity="1"/>
    <ColorMapEntry color="#3ECC1B" quantity="300" label="label" opacity="1"/>
    <ColorMapEntry color="#886363" quantity="350" label="label" opacity="1"/>
    <ColorMapEntry color="#5194CC" quantity="400" label="label" opacity="1"/>
    <ColorMapEntry color="#2C58DD" quantity="450" label="label" opacity="1"/>
    <ColorMapEntry color="#DDB02C" quantity="600" label="label" opacity="1"/>
</ColorMap>
```



The 'intervals' value means that every interval defined by two entries will be colorized using the value of the first entry, i.e. no color interpolation is applied between the intervals:

```
<ColorMap type="intervals" extended="true">
    <ColorMapEntry color="#EEBE2F" quantity="-300" label="label" opacity="0"/>
    ...
    <ColorMapEntry color="#DDB02C" quantity="600" label="label" opacity="1"/>
</ColorMap>
```

## Une réponse efficace à la directive INSPIRE

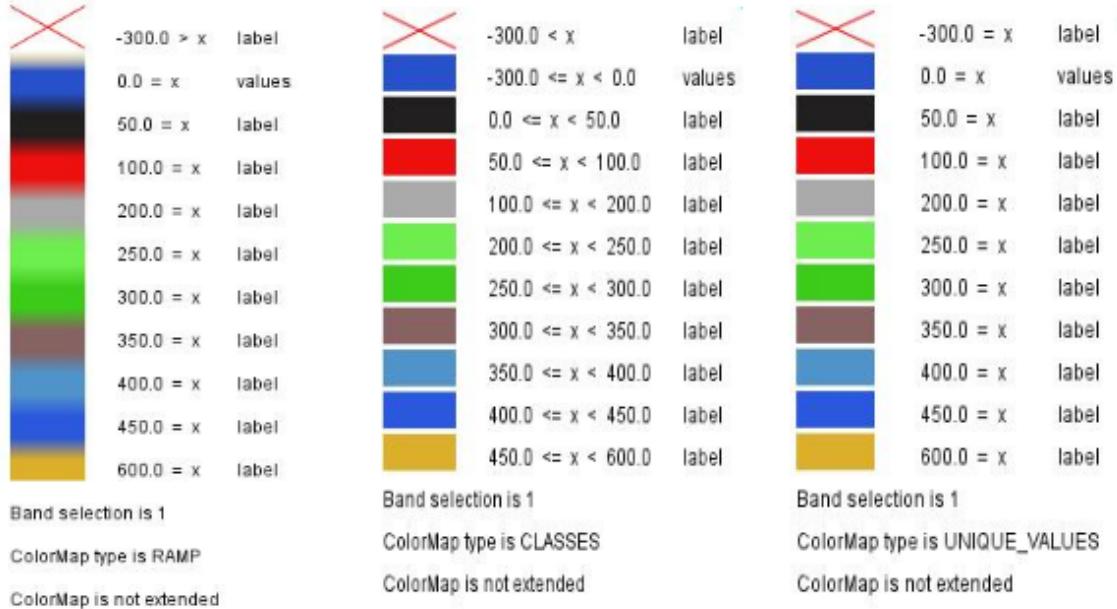


The 'extended' attribute allows ColorMap to compute gradients using 256 or 65536 colors; extended=false means that the color scale is calculated on 8 bit, else 16 bit if the value is true.

The difference between ramp, values and intervals values is also visible into raster legend. In order to get the raster legend from GeoServer the typically request is:

```
http://localhost:8080/geoserver/wms?REQUEST=GetLegendGraphic&VERSION=1.0.0&&STYLE=raster
```

the results are:

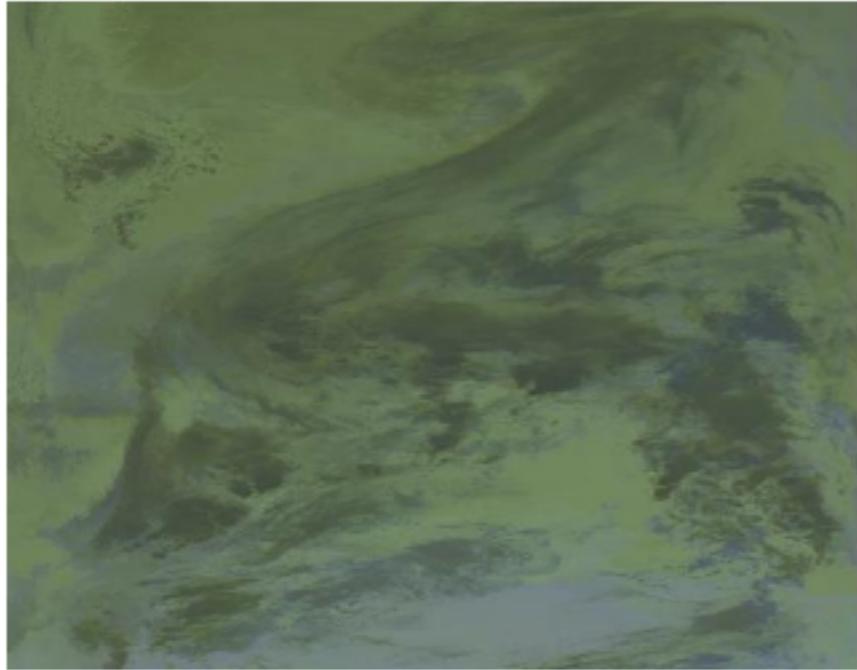


### ChannelSelection

This element specifies which color channel to access in the dataset. A dataset may contain standard three-channel colors (red, green, and blue channels) or one grayscale channel. Using `<ChannelSelection>` allows the mapping of a dataset channel to either a red, green, blue, or gray channel:

Une réponse efficace à la directive INSPIRE

```
<ChannelSelection>
  <RedChannel>
    <SourceChannelName>1</SourceChannelName>
  </RedChannel>
  <GreenChannel>
    <SourceChannelName>2</SourceChannelName>
  </GreenChannel>
  <BlueChannel>
    <SourceChannelName>3</SourceChannelName>
  </BlueChannel>
</ChannelSelection>
```

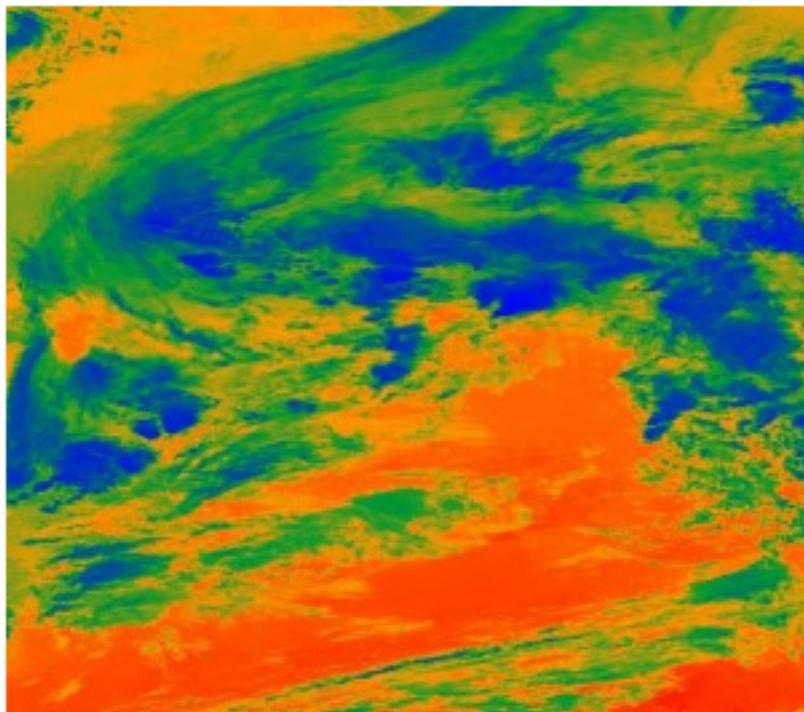


The above would map source channels 1, 2, and 3 to the red, green, and blue Channels, respectively.

This is the result of gray ChannelSelection operation applied to an RGB image and re-colorized through a ColorMap:

```
<RasterSymbolizer>
  <Opacity>1.0</Opacity>
  <ChannelSelection>
    <GrayChannel>
      <SourceChannelName>11</SourceChannelName>
    </GrayChannel>
  </ChannelSelection>
  <ColorMap extended="true">
    <ColorMapEntry color="#0000ff" quantity="3189.0"/>
    <ColorMapEntry color="#009933" quantity="6000.0"/>
    <ColorMapEntry color="#ff9900" quantity="9000.0" />
    <ColorMapEntry color="#ff0000" quantity="14265.0"/>
  </ColorMap>
```

```
</RasterSymbolizer>
```



### **ContrastEnhancement**

The `<ContrastEnhancement>` element is used in color channels to adjust the relative brightness of the data in that channel. There are three types of enhancements possible.

- Normalize
- Histogram
- GammaValue

Normalize means to expand the contrast so that the minimum quantity is mapped to minimum brightness, and the maximum quantity is mapped to maximum brightness. Histogram is similar to Normalize, but the algorithm used attempts to produce an image with an equal number of pixels at all brightness levels. Finally, GammaValue is a scaling factor that adjusts the brightness of the data, with a value less than one (1) darkening the image, and a value greater than one (1) brightening it. (Normalize and Histogram do not have any parameters.) One can use `<ContrastEnhancement>` on a specific channel (say red only) as opposed to globally, if it is desired. In this way, different enhancements can be used on each channel:

```
<ContrastEnhancement>
  <Normalize/>
</ContrastEnhancement>
```

```
<ContrastEnhancement>
  <Histogram/>
</ContrastEnhancement>
```

Une réponse efficace à la directive INSPIRE

These examples turn on Normalize and Histogram, respectively:

```
<ContrastEnhancement>
  <GammaValue>2</GammaValue>
</ContrastEnhancement>
```

The above increases the brightness of the data by a factor of two.

### ***ShadedRelief***

#### **Avis**

Support for this elements has not been implemented yet.

The `<ShadedRelief>` element can be used to create a 3-D effect, by selectively adjusting brightness. This is a nice effect to use on an elevation dataset. There are two types of shaded relief possible.

- BrightnessOnly
- ReliefFactor

BrightnessOnly, which takes no parameters, applies shading in WHAT WAY? ReliefFactor sets the amount of exaggeration of the shading (for example, to make hills appear higher). According to the OGC SLD specification, a value of around 55 gives "reasonable results" for Earth-based datasets:

```
<ShadedRelief>
  <BrightnessOnly />
  <ReliefFactor>55</ReliefFactor>
</ShadedRelief>
```

The above example turns on Relief shading in WHAT WAY?

### ***OverlapBehavior***

#### **Avis**

Support for this elements has not been implemented yet.

Sometimes raster data is comprised of multiple image sets. Take, for example, a [satellite view of the Earth at night](#). As all of the Earth can't be in nighttime at once, a composite of multiple images are taken. These images are georeferenced, and pieced together to make the finished product. That said, it is possible that two images from the same dataset could overlap slightly, and the OverlapBehavior element is designed to determine how this is handled. There are four types of OverlapBehavior:

- AVERAGE

Une réponse efficace à la directive INSPIRE

- RANDOM
- LATEST\_ON\_TOP
- EARLIEST\_ON\_TOP

**AVERAGE** takes each overlapping point and displays their average value. **RANDOM** determines which image gets displayed according to chance (which can sometimes result in a crisper image). **LATEST\_ON\_TOP** and **EARLIEST\_ON\_TOP** sets the determining factor to be the internal timestamp on each image in the dataset. None of these elements have any parameters, and are all called in the same way:

```
<OverlapBehavior>
    <AVERAGE />
</OverlapBehavior>
```

The above sets the OverlapBehavior to AVERAGE.

### ***ImageOutline***

#### **Avis**

Support for this elements has not been implemented yet.

Given the situation mentioned previously of the image composite, it is possible to style each image so as to have an outline. One can even set a fill color and opacity of each image; a reason to do this would be to "gray-out" an image. To use ImageOutline, you would define a **<LineSymbolizer>** or **<PolygonSymbolizer>** inside of the element:

```
<ImageOutline>
    <LineSymbolizer>
        <Stroke>
            <CssParameter name="stroke">#0000ff</CssParameter>
        </Stroke>
    </LineSymbolizer>
</ImageOutline>
```

The above would create a border line (colored blue with a one pixel default thickness) around each image in the dataset.

### ***TextSymbolizer***

The TextSymbolizer specifies **text labels**.

#### **Syntax**

A **<TextSymbolizer>** contains the following tags:

Tag	Required?	Description
<b>&lt;Label&gt;</b>	Yes	Specifies the content of the text label
<b>&lt;Font&gt;</b>	No	Specifies the font information for the labels.

Une réponse efficace à la directive INSPIRE

<LabelPlacement>	No	Sets the position of the label relative its associate feature.
<Halo>	No	Creates a colored background around the text label, for low contrast situations.
<Fill>	No	Determines the fill color of the text label.

Each of the above tags have additional sub tags. For the <Label> tag:

Tag	Required?	Description

Within the <Font> tag, the <CssParameter> tag is the only tag included. There are four types of parameters for this tag, each included inside the <CssParameter> tag:

Parameter	Required?	Description
name="font-family"	No	Determines the family name of the font to use for the label. Default is Times.
name="font-style"	No	Determines the style of the font. Options are normal, italic, and oblique. Default is normal.
name="font-weight"	No	Determines the weight of the font. Options are normal and bold. Default is normal.
name="font-size"	No	Determines the size of the font in pixels. Default is 10.

Within the <LabelPlacement> tag, there are many tags for specifying the placement of the label:

Tag	Required?	Description

Within the <Halo> tag, there are two tags to specify the details of the halo:

Tag	Required?	Description
<Radius>	No	Sets the size of the halo radius in pixels. Default is 1.
<Fill>	No	Sets the color of the halo in the form of #RRGGBB. Default is white (#FFFFFF).

The <Fill> tag is identical to that described in the WHERE~? above.

### **Example**

### **Labeling**

#### **Controlling Label Placement**

Controlling where the WMS server places labels with SLD is bit complex. The SLD specification only defines the most basic way of controlling placement explicitly says that defining more control is "a real can of worms". Geoserver fully supports the SLD specification plus adds a few extra parameters so you can make pretty maps.

#### **Basic SLD Placement**

The SLD specification indicates two types of LabelPlacement:

## Une réponse efficace à la directive INSPIRE

- for Point Geometries ("PointPlacement")
- for Linear (line) geometries ("LinePlacement")

### Note

Relative to Where?

**See below for the actual algorithm details, but:**

- Polygons are intersected with the viewport and the centroid is used.
- Lines are intersected with the viewport and the middle of the line is used.

### Code

```
<xsd:element name="PointPlacement">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="sld:AnchorPoint" minOccurs="0"/>
      <xsd:element ref="sld:Displacement" minOccurs="0"/>
      <xsd:element ref="sld:Rotation" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
...
<xsd:element name="LinePlacement">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="sld:PerpendicularOffset" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

### PointPlacement

When you use a <PointPlacement> element, the geometry you are labeling will be reduced to a single point (usually the "middle" of the geometry - see algorithm below for details). You can control where the label is relative to this point using the options:

Option	Meaning (Name)
AnchorPoint	This is relative to the LABEL. Using this you can do things such as center the label on top of the point, have the label to the left of the point, or have the label centered under the point.
Displacement	This is in PIXELS and lets you fine-tune the location of the label.
Rotation	This is the clockwise rotation of the label in degrees.

The best way to understand these is with examples:

### AnchorPoint

## Une réponse efficace à la directive INSPIRE

The anchor point determines where the label is placed relative to the label point. These measurements are relative to the bounding box of the label. The (x,y) location inside the label's bounding box (specified by the AnchorPoint) is placed at the label point.



The anchor point is defined relative to the label's bounding box. The bottom left is (0,0), the top left is (1,1), and the middle is (0.5,0.5).

```
<PointPlacement>
  <AnchorPoint>
    <AnchorPointX>
      0.5
    </AnchorPointX>
    <AnchorPointY>
      0.5
    </AnchorPointY>
  </AnchorPoint>
</PointPlacement>
```

By changing the values, you can control where the label is placed.



(x=0,y=0.5) DEFAULT - place the label to the right of the label point



(x=0.5,y=0.5) - place the centre of the label at the label point



(x=1,y=0.5) - place the label to the left of the label point



(x=0.5,y=0) - place the label centered above the label point

### **Displacement**

Displacement allows fine control of the placement of the label. The displacement values are in pixels and simply move the location of the label on the resulting image.

```
<PointPlacement>
  <Displacement>
    <DisplacementX>
      10
    </DisplacementX>
    <DisplacementY>
      0
    </DisplacementY>
  </Displacement>
</PointPlacement>
```

Une réponse efficace à la directive INSPIRE



displacement of x=10 pixels, compare with anchor point (x=0,y=0.5) above



displacement of y=-10 pixels, compare with anchor point (x=0.5,y=1.0) not shown

### **Rotation**

Rotation is simple - it rotates the label clockwise the number of degrees you specify. See the examples below for how it interacts with AnchorPoints and displacements.

```
<Rotation>
  45
</Rotation>
```



simple 45 degrees rotation



45 degrees rotation with anchor point (x=0.5,y=0.5)



45 degrees with 40 pixel X displacement



45 degrees rotation with 40 pixel Y displacement with anchor point (x=0.5,y=0.5)

### **LinePlacement**

When you are labeling a line (i.e. a road or river), you can specify a `<LinePlacement>` element. This tells the labeling system two things: (a) that you want Geoserver to determine the best rotation and placement for the label (b) a minor option to control how the label is placed relative to the line.

The line placement option is very simple - it only allows you to move a label up-and-down from a line.

```
<xs:elementname="LinePlacement">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sld:PerpendicularOffset" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

Une réponse efficace à la directive INSPIRE

```
</xs:element>
...
<xs:element name="PerpendicularOffset" type="sld:ParameterValueType"/>
```

This is very similar to the DisplacementY option (see above).

```
<LabelPlacement>
  <LinePlacement>
    <PerpendicularOffset>
      10
    </PerpendicularOffset>
  </LinePlacement>
</LabelPlacement>
```



PerpendicularOffset=0



PerpendicularOffset=10 pixels

### **Composing labels from multiple attributes**

The `<Label>` element in TextSymbolizer is said to be mixed, that is, its content can be a mixture of plain text and OGC Expressions. The mix gets interpreted as a concatenation, this means you can leverage it to get complex labels out of multiple attributes.

For example, if you want both a state name and its abbreviation to appear in a label, you can do the following:

```
<Label>
  <ogc:PropertyName>STATE_NAME</ogc:PropertyName> (<ogc:PropertyName>STATE_ABBR</ogc:PropertyName>)
</Label>
```

and you'll get a label such as **Texas (TX)**.

If you need to add extra white space or newline, you'll stumble into an XML oddity. The whitespace handling in the Label element is following a XML mandated rule called "collapse", in which all leading and trailing whitespaces have to be removed, whilst all whitespaces (and newlines) in the middle of the xml element are collapsed into a single whitespace.

So, what if you need to insert a newline or a sequence of two or more spaces between your property names? Enter CDATA. CDATA is a special XML section that has to be returned to the interpreter as-is, without following any whitespace handling rule. So, for example, if you wanted to have the state abbreviation sitting on the next line you'd use the following:

Une réponse efficace à la directive INSPIRE

```
<Label>
  <ogc:PropertyName>STATE_NAME</ogc:PropertyName><![CDATA[
  ]]>(<ogc:PropertyName>STATE_ABBR</ogc:PropertyName>)
</Label>
```

### Geoserver Specific Enhanced Options

The following options are all extensions of the SLD specification. Using these options gives much more control over how the map looks, since the SLD standard isn't expressive enough to handle all the options one might want. In time we hope to have them be an official part of the specification.

#### Priority Labeling (<Priority>)

GeoServer has extended the standard SLD to also include priority labeling. This allows you to control which labels are rendered in preference to other labels.

For example, lets assume you have a data set like this:

City Name	population
Yonkers	197,818
Jersey City	237,681
Newark	280,123
New York	8,107,916

Most people don't know where "Yonkers" city is, but do know where "New York" city is. On our map, we want to give "New York" priority so its more likely to be labeled when it's in conflict (overlapping) "Yonkers".

#### Note

##### Standard SLD Behavior

If you do not have a <Priority> tag in your SLD then you get the default SLD labeling behavior. This basically means that if there's a conflict between two labels, there is no 'dispute' mechanism and its random which label will be displayed.

In our TextSymbolizer we can put an Expression to retrieve or calculate the priority for each feature:

```
<Priority>
  <PropertyName>population</PropertyName>
</Priority>
```



Location of the cities (see population data above)



## Une réponse efficace à la directive INSPIRE

New York is labeled in preference to the less populated cities. Without priority labeling, "Yonkers" could be labeled in preference to New York, making a difficult to interpret map.

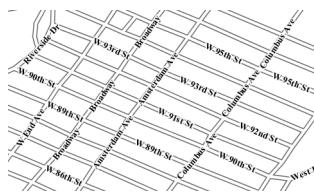


Notice that larger cities are more readily named than smaller cities.

### **Grouping Geometries (<VendorOption name="group">)**

Sometimes you will have a set of related features that you only want a single label for. The grouping option groups all features with the same label text, then finds a representative geometry for the group.

Roads data is an obvious example - you only want a single label for all of "main street", not a label for every piece of "main street"."



When the grouping option is off (default), grouping is not performed and each geometry is labeled (space permitting).



With the grouping option on, all the geometries with the same label are grouped together and the label position is determined from ALL the geometries.

Geometry	Representative Geometry
Point Set	first point inside the view rectangle is used.
Line Set	lines are (a) networked together (b) clipped to the view rectangle (c) middle of the longest network path is used.
Polygon Set	polygons are (a) clipped to the view rectangle (b) the centroid of the largest polygon is used.

```
<VendorOption name="group">yes</VendorOption>
```

## Avis

Watch out - you could group together two sets of features by accident. For example, you could create a single group for "Paris" which contains features for Paris (France) and Paris (Texas).

### **Overlapping and Separating Labels (<VendorOption name="spaceAround">)**

By default geoserver will not put labels "on top of each other". By using the spaceAround option you can allow labels to overlap and you can also add extra space around a label.

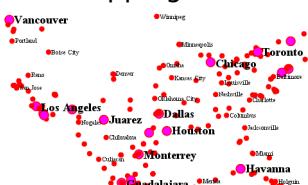
```
<VendorOption name="spaceAround">10</VendorOption>
```



Default behavior ("0") - the bounding box of a label cannot overlap the bounding box of another label.



With a negative spaceAround value, overlapping is allowed.



With a spaceAround value of 10 for all TextSymbolizers, each label will be 20 pixels apart from each other (see below).

**NOTE:** the value you specify (an integer in pixels) actually provides twice the space that you might expect. This is because you can specify a spaceAround for one label as 5, and for another label (in another TextSymbolizer) as 3. The distance between them will be 8. For two labels in the first symbolizer ("5") they will each be 5 pixels apart from each other, for a total of 10 pixels!

## Note

### **Interaction with different values in different TextSymbolizers**

You can have multiple TextSymbolizers in your SLD file, each with a different spaceAround option. This will normally do what you would think if all your

Une réponse efficace à la directive INSPIRE

spaceAround options are  $\geq 0$ . If you have negative values ('allow overlap') then these labels can overlap labels that you've said should not be overlapping. If you don't like this behavior, it's not too difficult to change - feel free to submit a patch!

### **followLine**

The **followLine** option forces a label to follow the curve of the line. To use this option place the following in your `<TextSymbolizer>`.

```
<VendorOption name="followLine">true</VendorOption>
```

It is required to use `<LinePlacement>` along with this option to ensure that all labels are correctly following the lines:

```
<LabelPlacement>
  <LinePlacement/>
</LabelPlacement>
```

### **maxDisplacement**

The **maxDisplacement** option controls the displacement of the label along a line. Normally GeoServer would label a line at its center point only, provided the location is not busy with another label, and not label it at all otherwise. When set, the labeller will search for another location within **maxDisplacement** pixels from the pre-computed label point.

When used in conjunction with **repeat**, the value for **maxDisplacement** should always be lower than the value for repeat.

```
<VendorOption name="maxDisplacement">10</VendorOption>
```

### **repeat**

The **repeat** option determines how often GeoServer labels a line. Normally GeoServer would label each line only once, regardless of their length. Specify a positive value to make it draw the label every **repeat** pixels.

```
<VendorOption name="repeat">100</VendorOption>
```

### **labelAllGroup**

The **labelAllGroup** option makes sure that all of the segments in a line group are labeled instead of just the longest one.

```
<VendorOption name="labelAllGroup">true</VendorOption>
```

### **maxAngleDelta**

Designed to be used in conjunction with **followLine**, the **maxAngleDelta** option sets the maximum angle, in degrees, between two subsequent characters in a curved label. Large angles create either visually disconnected words or overlapping characters. It is advised not to use angles larger than 30.

Une réponse efficace à la directive INSPIRE

```
<VendorOption name="maxAngleDelta">15</VendorOption>
```

### **autoWrap**

The **autoWrap** option wraps labels when they exceed the given value, given in pixels. Make sure to give a dimension wide enough to accommodate the longest word other wise this option will split words over multiple lines.

```
<VendorOption name="autoWrap">50</VendorOption>
```

### **forceLeftToRight**

The labeller always tries to draw labels so that they can be read, meaning the label does not always follow the line orientation, but sometimes it's flipped 180° instead to allow for normal reading. This may get in the way if the label is a directional arrow, and you're trying to show one way directions (assuming the geometry is oriented along the one way, and that you have a flag to discern one ways from streets with both circulations).

The following setting disables label flipping, making the label always follow the natural orientation of the line being labelled:

```
<VendorOption name="forceLeftToRigth">false</VendorOption>
```

### **conflictResolution**

By default labels are subjected to conflict resolution, meaning the renderer will not allow any label to overlap with a label that has been drawn already. Setting this parameter to false pull the label out of the conflict resolution game, meaning the label will be drawn even if it overlaps with other labels, and other labels drawn after it won't mind overlapping with it.

```
<VendorOption name="conflictResolution">false</VendorOption>
```

### **Goodness of Fit**

Geoserver will remove labels if they are a particularly bad fit for the geometry they are labeling.

Geometry	Goodness of Fit Algorithm
Point	Always returns 1.0 since the label is at the point
Line	Always returns 1.0 since the label is always placed on the line.
Polygon	The label is sampled approximately at every letter. The distance from these points to the polygon is determined and each sample votes based on how close it is to the polygon. (see LabelCacheDefault#goodnessOfFit())

The default value is 0.5, but it can be modified using:

```
<VendorOption name="goodnessOfFit">0.3</VendorOption>
```

### **Polygon alignment**

GeoServer normally tries to place horizontal labels within a polygon, and give up in case the label position is busy or if the label does not fit enough in the polygon. This options allows

Une réponse efficace à la directive INSPIRE

GeoServer to try alternate rotations for the labels.

```
<VendorOption name="polygonAlign">mbr</VendorOption>
```

Option	Description
manual	The default value, only the rotation manually specified in the <Rotation> tag will be used
ortho	If the label does not fit horizontally and the polygon is taller than wider the vertical alignment will also be tried
mbr	If the label does not fit horizontally the minimum bounding rectangle will be computed and a label aligned to it will be tried out as well

### **Filters**

A *filter* is the mechanism in SLD for specifying predicates. Similar in nature to a "WHERE" clause in SQL, filters are the language for specifying which styles should be applied to which features in a data set.

The filter language used by SLD is itself an [OGC standard](#) defined in the Filter Encoding specification freely available.

A filter is used to select a subset of features of a dataset to apply a symbolizer to.

There are three types of filters:

#### **Attribute filters**

Attribute filters are used to constrain the non-spatial attributes of a feature. Example

```
1 <PropertyIsEqualTo>
2   <PropertyName>NAME</PropertyName>
3   <Literal>Bob</Literal>
4 </PropertyIsEqualTo>
```

The above filter selects those features which have a {{NAME}} attribute which has a value of "Bob". A variety of equality operators are available:

- PropertyIsEqualTo
- PropertyIsNotEqualTo
- PropertyIsLessThan
- PropertyIsLessThanOrEqualTo
- PropertyIsGreaterThan
- PropertyIsGreaterThanOrEqualTo
- PropertyIsBetween

#### **Spatial filters**

Spatial filters used to constrain the spatial attributes of a feature. Example

Une réponse efficace à la directive INSPIRE

```
1 <Intersects>
2   <PropertyName>GEOMETRY</PropertyName>
3   <Literal>
4     <gml:Point>
5       <gml:coordinates>1 1</gml:coordinates>
6     </gml:Point>
7   </Literal>
8 </Intersects>
```

The above filter selects those features with a geometry that intersects the point (1,1). A variety of spatial operators are available:

- Intersects
- Equals
- Disjoint
- Within
- Overlaps
- Crosses
- DWithin
- Beyond
- Distance

### ***Logical filters***

Logical filters are used to create combinations of filters using the logical operators And, Or, and Not. Example

```
1 <And>
2   <PropertyIsEqualTo>
3     <PropertyName>NAME</PropertyName>
4     <Literal>Bob</Literal>
5   </PropertyIsEqualTo>
6   <Intersects>
7     <PropertyName>GEOMETRY</PropertyName>
8     <Literal>
9       <gml:Point>
10      <gml:coordinates>1 1</gml:coordinates>
11      </gml:Point>
12    </Literal>
13  </Intersects>
14 </And>
```

### ***Rules***

A rule combines a number of symbolizers with a filter to define the portrayal of a feature. Consider the following example:

```
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsGreaterThan>
```

Une réponse efficace à la directive INSPIRE

```
<ogc:PropertyName>POPULATION</ogc:PropertyName>
  <ogc:Literal>100000</ogc:Literal>
</ogc:PropertyIsGreaterThan>
</ogc:Filter>
<PointSymbolizer>
  <Graphic>
    <Mark>
      <Fill><CssParameter name="fill">#FF0000</CssParameter>
    </Mark>
  </Graphic>
</PointSymbolizer>
</Rule>
```

The above rule applies only to features which have a POPULATION attribute greater than 100,000 and symbolizes them with a red point.

An SLD document can contain many rules. Multiple rule SLD's are the basis for "thematic styling". Consider the above example expanded:

```
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsGreaterThan>
      <ogc:PropertyName>POPULATION</ogc:PropertyName>
      <ogc:Literal>100000</ogc:Literal>
    </ogc:PropertyIsGreaterThan>
  </ogc:Filter>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <Fill><CssParameter name="fill">#FF0000</CssParameter>
      </Mark>
    </Graphic>
  </PointSymbolizer>
</Rule>
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsLessThan>
      <ogc:PropertyName>POPULATION</ogc:PropertyName>
      <ogc:Literal>100000</ogc:Literal>
    </ogc:PropertyIsLessThan>
  </ogc:Filter>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <Fill><CssParameter name="fill">#0000FF</CssParameter>
      </Mark>
    </Graphic>
  </PointSymbolizer>
</Rule>
```

The above snippet defines an additional rule which engages when POPULATION is less than 100,000 and symbolizes the feature as a green point.

Rules support the notion of *scale dependence* which allows one to specify the scale at which a rule should engage. This allows for different portrayals of a feature based on map scale.

Une réponse efficace à la directive INSPIRE

Consider the following example:

```
<Rule>
  <MaxScaleDenominator>20000</MaxScaleDenominator>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <Fill><CssParameter name="fill">#FF0000</CssParameter>
      </Mark>
    </Graphic>
  </PointSymbolizer>
</Rule>
<Rule>
  <MinScaleDenominator>20000</MinScaleDenominator>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <Fill><CssParameter name="fill">#0000FF</CssParameter>
      </Mark>
    </Graphic>
  </PointSymbolizer>
</Rule>
```

The above rules specify that at a scale below 1:20000 features are symbolized with red points, and at a scale above 1:20000 features are symbolized with blue points.

### **Scale**

#### **SLD Extensions in GeoServer**

GeoServer sports a number of vendor specific extensions to SLD 1.0. While not portable they allow to make map making more flexible and to generate better looking maps.

#### **Transformations de géométries en SLD**

En SLD 1.0, chaque symboliseur contient un élément `<Geometry>` qui permet à l'utilisateur de spécifier quelle géométrie employer pour le rendu. Dans la plupart des cas ce n'est pas utilisé, mais cela peut devenir utile lorsqu'un objet comprend plusieurs géométries.

SLD 1.0 exige que le contenu de `<Geometry>` soit une propriété `<ogc:PropertyName>`, GeoServer assouplit cette contrainte et permet d'utiliser à la place une expression générique `sld:expression`. Les expressions usuelles ne savent pas opérer sur les géométries, mais GeoServer fournit un lot de fonctions filtres qui peuvent effectivement manipuler les géométries en les transformant en quelque chose de différent: nous appelons cela *transformations de géométries* en SLD.

Une liste complète des transformations disponibles est disponible dans `filter_function_reference`.

Les transformations sont plutôt souples, leur principale limitation est qu'elles interviennent dans le système de référence et les unitées propres à la géométrie, avant toute reprojection et rééchantillonnage pour affichage.

Voyons quelques exemples.

#### **Extraction des sommets**

## Une réponse efficace à la directive INSPIRE

Voici un exemple qui permet d'extraire tous les sommets d'une géométrie pour les rendre visibles sur une carte, en utilisant la fonction `vertices`:

```
1  <PointSymbolizer>
2    <Geometry>
3      <ogc:Function name="vertices">
4        <ogc:PropertyName>the_geom</ogc:PropertyName>
5      </ogc:Function>
6    </Geometry>
7    <Graphic>
8      <Mark>
9        <WellKnownName>square</WellKnownName>
10       <Fill>
11         <CssParameter name="fill">#FF0000</CssParameter>
12       </Fill>
13     </Mark>
14     <Size>6</Size>
15   </Graphic>
16 </PointSymbolizer>
```

Consultez et téléchargez le SLD complet "Sommets"

Appliqué à la couche exemple `tasmania_roads`, ceci aboutit à:



*Extraire et afficher les sommets extraits d'une géométrie*

### Points début et fin

Les fonctions `startPoint` et `endPoint` peuvent être utilisées pour extraire les points début et fin d'une ligne.

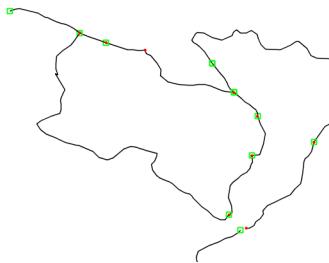
```
1 <PointSymbolizer>
2   <Geometry>
3     <ogc:Function name="startPoint">
4       <ogc:PropertyName>the_geom</ogc:PropertyName>
5     </ogc:Function>
6   </Geometry>
7   <Graphic>
8     <Mark>
9       <WellKnownName>square</WellKnownName>
10      <Stroke>
11        <CssParameter name="stroke">0x00FF00</CssParameter>
12        <CssParameter name="stroke-width">1.5</CssParameter>
13      </Stroke>
14    </Mark>
15    <Size>8</Size>
```

Une réponse efficace à la directive INSPIRE

```
16  </Graphic>
17 </PointSymbolizer>
18 <PointSymbolizer>
19   <Geometry>
20     <ogc:Function name="endPoint">
21       <ogc:PropertyName>the_geom</ogc:PropertyName>
22     </ogc:Function>
23   </Geometry>
24   <Graphic>
25     <Mark>
26       <WellKnownName>circle</WellKnownName>
27       <Fill>
28         <CssParameter name="fill">0xFF0000</CssParameter>
29       </Fill>
30     </Mark>
31     <Size>4</Size>
32   </Graphic>
33 </PointSymbolizer>
```

Consultez et téléchargez le SLD complet "Début-Fin"

Appliqué à la couche exemple *tasmania\_roads*, ceci aboutit à:



*Extraire les points début et fin d'une ligne*

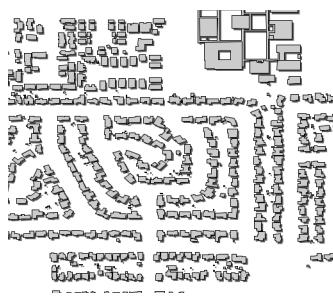
### Ombre portée

La fonction *offset* peut être utilisée pour créer un effet d'ombre portée derrière les polygones. Notez la valeur bizarre du décalage, choisie ainsi car les données utilisées dans l'exemple sont exprimées en coordonnées géographiques.

```
1  <PolygonSymbolizer>
2    <Geometry>
3      <ogc:Function name="offset">
4        <ogc:PropertyName>the_geom</ogc:PropertyName>
5        <ogc:Literal>0.00004</ogc:Literal>
6        <ogc:Literal>-0.00004</ogc:Literal>
7      </ogc:Function>
8    </Geometry>
9    <Fill>
10      <CssParameter name="fill">#555555</CssParameter>
11    </Fill>
12  </PolygonSymbolizer>
```

Consultez et téléchargez le SLD complet "Ombre"

Appliqué à la couche exemple *tasmania\_roads*, ceci aboutit à:



Ombre portée

### **Autres possibilités**

L'assortiment de fonctions de transformations dans GeoServer contient aussi un lot de transformations relatives ou constructives comme buffer, intersection, différence et ainsi de suite. Toutefois, ces fonctions sont relativement exigeantes en terme de consommation CPU, il est donc conseillé de les utiliser avec précaution et de ne les activer qu'à de hauts niveaux de zoom.

Les buffer peuvent souvent être approchés en choisissant des traits très larges et des jointures de lignes ronds, sans nécessiter des transformations géométriques.

### **Ajouter de nouvelles transformations**

Les fonctions filtre sont pluggable, ce qui signifie qu'il est possible d'en construire de nouvelles en Java et de déposer le fichier .jar résultat dans GeoServer en tant que plugin. A l'heure actuelle aucun guide n'est disponible, mais vous pouvez jeter un oeil dans le module principal GeoTools pour trouver des exemples.

### **Point symbology in GeoServer**

Point symbology in GeoServer is supported via the SLD Graphic element. The element can appear in PointSymbolizer, but also be used as a GraphicStroke to repeat symbols over lines and as a GraphicFill to fill polygons with tiled and repeated symbols.

Graphic can either contain a Mark or a ExternalGraphic. Marks are simple vector symbols that can be stroked and filled at the SLD editor command, whilst ExternalGraphic refer to external files, such as PNG or SVG, that already contain all the information needed to paint the symbol, both shape and color.

In SLD both Mark and ExternalGraphic are defined to be pretty static, however GeoServer allows the user to embed attribute names into them for value expansion at run-time, this is known as *dynamic symbolizer support*.

### **Mark support in GeoServer**

GeoServer supports all of the standard Mark defined in the SLD standard, plus an open ended set of symbol extensions. All the symbol names have to be placed in WellKnownName section of a Mark definition, see the [PointSymbolizer](#) reference for further details as well as the various examples in the cookbook section dedicated to [Points](#).

### **Built-in symbols**

The SLD specification mandates the support of the following symbols:

Name	Description
square	A square

circle	A circle
triangle	A triangle pointing up
star	five-pointed star
cross	A square cross with space around (not suitable for hatch fills)
x	A square X with space around (not suitable for hatch fills)

### **The shape symbols**

The shape symbols set adds a number of extra symbols that are not part of the basic set, they are all prefixed by shape://

Name	Description
shape://vertline	A vertical line (suitable for hatch fills or to make railroad symbols)
shape://horline	A horizontal line (suitable for hatch fills)
shape://slash	A diagonal line leaning forwards like the "slash" keyboard symbol (suitable for diagonal hatches)
shape://backslash	Same as shape://slash, but oriented in the opposite direction
shape://dot	A very small circle with space around
shape://plus	A + symbol, without space around (suitable for cross-hatch fills)
shape://times	A "X" symbol, without space around (suitable for cross-hatch fills)
shape://oarrow	An open arrow symbol (triangle without one side, suitable for placing arrows at the end of lines)
shape://carrow	An closed arrow symbol (closed triangle, suitable for placing arrows at the end of lines)

### **The TTF marks**

It is possible to create a mark out of any decorative/symbol True Type Font, such as Wingdings, WebDings, and the many symbol fonts available on the internet using the following syntax:

```
ttf://<fontname>#<hexcode>
```

where fontname is the full name of a TTF font available to GeoServer whilst hexcode is the hexadecimal code of the symbol. In order to get the hex code of a symbol it is possible to use the "char map" utilities available in major operating systems (Windows and Gnome on Linux both have one).

For example, say we want to use the "shield" symbol contained in Webdings, the Gnome charmap reports the symbol code as follows:

## Une réponse efficace à la directive INSPIRE

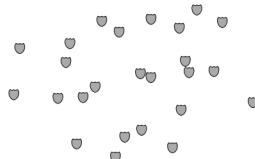


Selecting the code out of the Gnome char map

Thus the SLD snippet to use the shield will be:

```
1 <PointSymbolizer>
2   <Graphic>
3     <Mark>
4       <WellKnownName>ttf://Webdings#0x0064</WellKnownName>
5       <Fill>
6         <CssParameter name="fill">#AAAAAA</CssParameter>
7       </Fill>
8       <Stroke/>
9     </Mark>
10    <Size>16</Size>
11  </Graphic>
12 </PointSymbolizer>
```

This will result in the following map symbols:



Shields on the map

### **Adding your own**

The mark subsystem is open ended, one has just to implement the `MarkFactory` interface and declare its implementation in the `META-INF/services/org.geotools.renderer.style.MarkFactory` file.

While there is not much documentation the javadocs of the GeoTools `MarkFactory` along with the following example files should suffice:

- The factory SPI registration file
- The `TTFMarkFactory` implementation
- The `ShapeMarkFactory` implementation

### **External graphics in GeoServer**

## Une réponse efficace à la directive INSPIRE

ExternalGraphic is the other source of point symbology. Unlike marks these images are used as-is, so the specification is somewhat simpler, just point at the file and specify what type of file is that using its mime type:

```
1 <PointSymbolizer>
2   <Graphic>
3     <ExternalGraphic>
4       <OnlineResource xlink:type="simple" xlink:href="http://mywebsite.com/points.png"/>
5         <Format>image/png</Format>
6     </ExternalGraphic>
7   </Graphic>
8 </PointSymbolizer>
```

The size element can be specified as with the Mark, but when using raster graphic symbols it's better to avoid resizing them as that will blur them, and use them at their natural size instead.

The location of the symbol can also be a relative one, in that case the file will be searched inside \$GEOSERVER\_DATA\_DIR/styles, such as in the following example:

```
1 <PointSymbolizer>
2   <Graphic>
3     <ExternalGraphic>
4       <OnlineResource xlink:type="simple" xlink:href="burg02.svg" />
5         <Format>image/svg+xml</Format>
6     </ExternalGraphic>
7     <Size>20</Size>
8   </Graphic>
9 </PointSymbolizer>
```

In this particular example a SVG image has been used. SVG is a vector description having both shapes and color, as such it scales nicely at whatever size, thus using size is possible and recommended.

## **Dynamic symbolizers**

Both Mark well known name and ExternalGraphic/OnlineResource/href are supposed to be, in SLD, static strings, meaning they cannot change based on the current feature.

This makes for very verbose expressions when multiple symbols need to be used based on a feature attributes, as a different Rule and Symbolizer must be used for each different symbol.

GeoServer allows to embed attribute names, and indeed, any kind of valid CQL expression, inside both WellKnownName and OnlineResource/@xlink:href.

This allows for more compact styling assuming the name of the symbol to be used can be derived from the feature attribute values. For example, if we want to display the flags of the various states in the USA and assuming the file names match the state name the following style would allow to pick each and any of them using a single rule:

```
1 <ExternalGraphic>
2   <OnlineResource xlink:type="simple" xlink:href="http://mysite.com/tn_{STATE_ABBR}.gif"/>
3     <Format>image/gif</Format>
4 </ExternalGraphic>
```

## Une réponse efficace à la directive INSPIRE

If some adaptation to the name is necessary a full CQL expression can be used instead. In particular, if the name in the attribute is upper case but the URL demands a lowercase name the following could be used instead:

```
1 <ExternalGraphic>
2   <OnlineResource xlink:type="simple"
3     xlink:href="http://mysite.com/tn_{$strToLowercase(STATE_ABBR)}.jpg" />
4   <Format>image/gif</Format>
5 </ExternalGraphic>
```

Generally speaking any CQL expression can be embedded in a url or well known name by using the \${cql expression} syntax, where a simple attribute name such as \${STATE\_ABBR} is one of the simplest expression.

### **Parameter substitution in SLD**

Parameter substitution in SLD is a GeoServer specific functionality (starting from version 2.0.2) allowing to pass down parameter values from the WMS request into an SLD style, allowing to dynamically change colors, fonts, sizes and filter thresholds dynamically.

The parameters are specified on GetMap requests using the env parameter:

```
...&env=paramName:value;otherParam=otherValue&...
```

and can be accessed from the SLD using the env function. In the simplest form the env function will retrieve the value of a specific parameter:

```
<ogc:Function name="env">
  <ogc:Literal>size</ogc:Literal>
</ogc:Function>
```

Alternatively a default value can be provided, to be used if the parameter value was not provided along with the GetMap request:

```
<ogc:Function name="env">
  <ogc:Literal>size</ogc:Literal>
  <ogc:Literal>6</ogc:Literal>
</ogc:Function>
```

The function can be called in any place where an OGC expression is used, so for example CSSParameter, sizes and offsets, and filter themselves. The SLD parser accepts it even in some places where an expression is not formally accepted, as the mark well known names.

### **A working example**

The following symbolizer has been parametrized in three places, every time with fall backs (full SLD is also available):

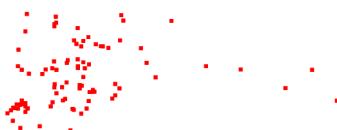
```
<PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName><ogc:Function name="env">
        <ogc:Literal>name</ogc:Literal>
        <ogc:Literal>square</ogc:Literal>
```

Une réponse efficace à la directive INSPIRE

```
</ogc:Function>
</WellKnownName>
<Fill>
  <CssParameter name="fill">
    #<ogc:Function name="env">
      <ogc:Literal>color</ogc:Literal>
      <ogc:Literal>FF0000</ogc:Literal>
    </ogc:Function>
  </CssParameter>
</Fill>
</Mark>
<Size>
  <ogc:Function name="env">
    <ogc:Literal>size</ogc:Literal>
    <ogc:Literal>6</ogc:Literal>
  </ogc:Function>
</Size>
</Graphic>
</PointSymbolizer>
```

Download the full SLD style

The SLD renders the sample sf:bugsites as follows when no parameter is provided in the request:



*Default rendering*

If the request is changed to include the following parameter:

```
env=color:00FF00;name:triangle;size:12
```

the result will be instead:



*Rendering with parameters*

### **Specifying symbolizers sizes in ground units**

The SLD 1.0 specification allows the specification of sizes in just one unit: pixels.

The Symbology Encoding 1.1 specification instead allows to use also meters and feet, as ground units, so that the size of the symbolizers changes on the screen as one zooms in and out.

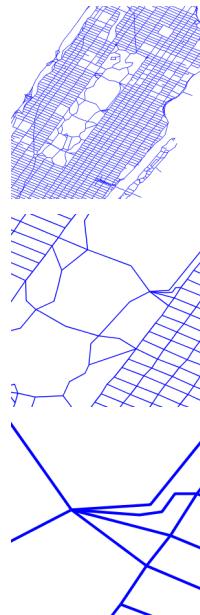
## Une réponse efficace à la directive INSPIRE

GeoServer supports the uom attribute just as specified in SE 1.1 in its extended SLD 1.0 support:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer>
    <Name>5m blue line</Name>
    <UserStyle>
      <Title>tm blue line</Title>
      <Abstract>Default line style, 5m wide blue</Abstract>

      <FeatureTypeStyle>
        <Rule>
          <Title>Blue Line, 5m large</Title>
          <LineSymbolizer uom="http://www.opengeospatial.org/se/units/metre">
            <Stroke>
              <CssParameter name="stroke">#0000FF</CssParameter>
              <CssParameter name="stroke-width">5</CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

Applying the style to tiger:tiger\_roads and zooming in we get:



The unit of measure supported are the same specified in the SE 1.1 specification:

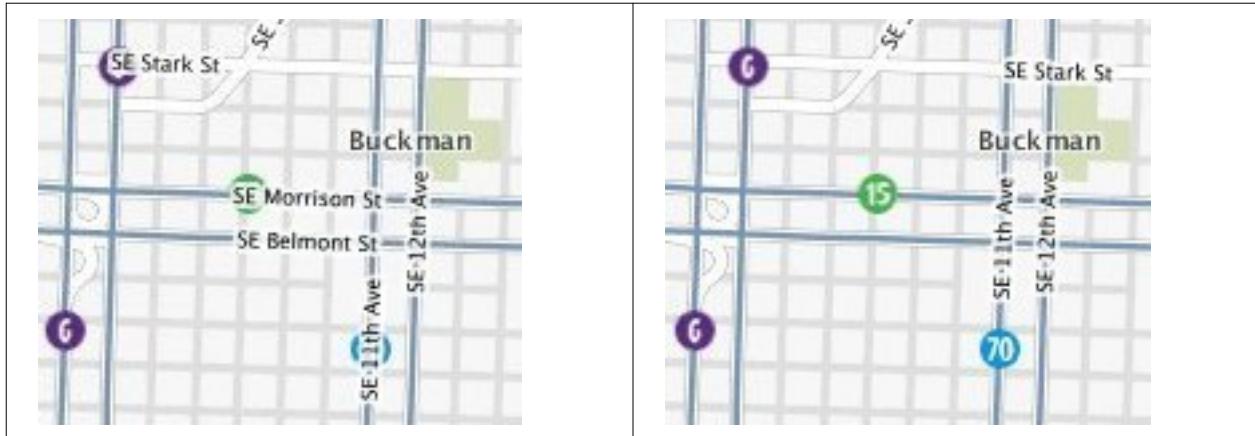
```
http://www.opengeospatial.org/se/units/metre
http://www.opengeospatial.org/se/units/foot
http://www.opengeospatial.org/se/units/pixel
```

## Une réponse efficace à la directive INSPIRE

This extended feature is officially supported starting with GeoServer 2.1.0, but it's already available in GeoServer 2.0.3 if the administrator starts the java virtual with the -DenableDpiUomRescaling=true system variable specification.

### **Label Obstacles**

GeoServer contains a number of internal algorithms for doing label conflict resolution that prevent labels from overlapping one another. However by default these algorithms only account for other labels. This can result in labels overlapping other symbolizers and that effect may be unintended.



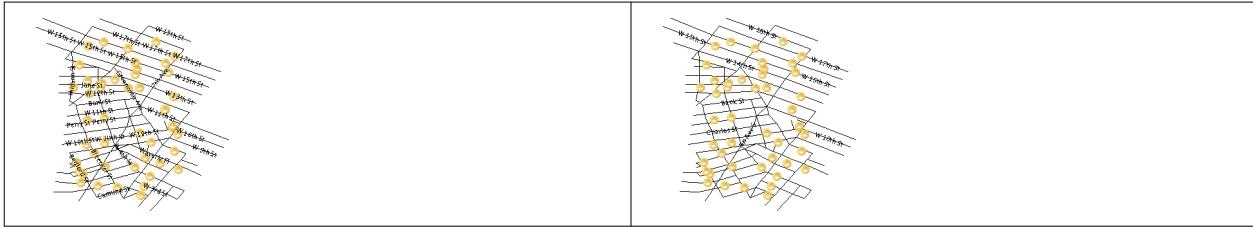
GeoServer supports a vendor option called `labelObstacle` that allows one to mark a symbolizer as an obstacle that tells the labeller to avoid rendering a label that overlaps with it.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer>
    <UserStyle>

      <FeatureTypeStyle>
        <Rule>
          <PointSymbolizer>
            <Graphic>
              <ExternalGraphic>
                <OnlineResource
                  xlink:type="simple"
                  xlink:href="smileyface.png" />
                <Format>image/png</Format>
              </ExternalGraphic>
              <Size>32</Size>
            </Graphic>
            <VendorOption name="labelObstacle">true</VendorOption>
          </PointSymbolizer>
        </Rule>
      </FeatureTypeStyle>

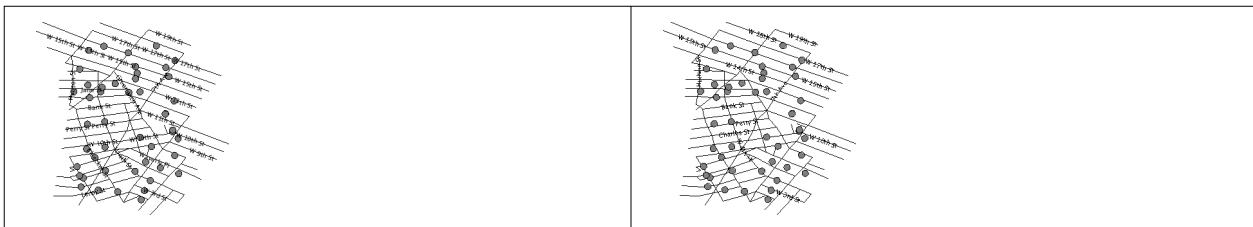
      </UserStyle>
    </NamedLayer>
  </StyledLayerDescriptor>
```

## Une réponse efficace à la directive INSPIRE

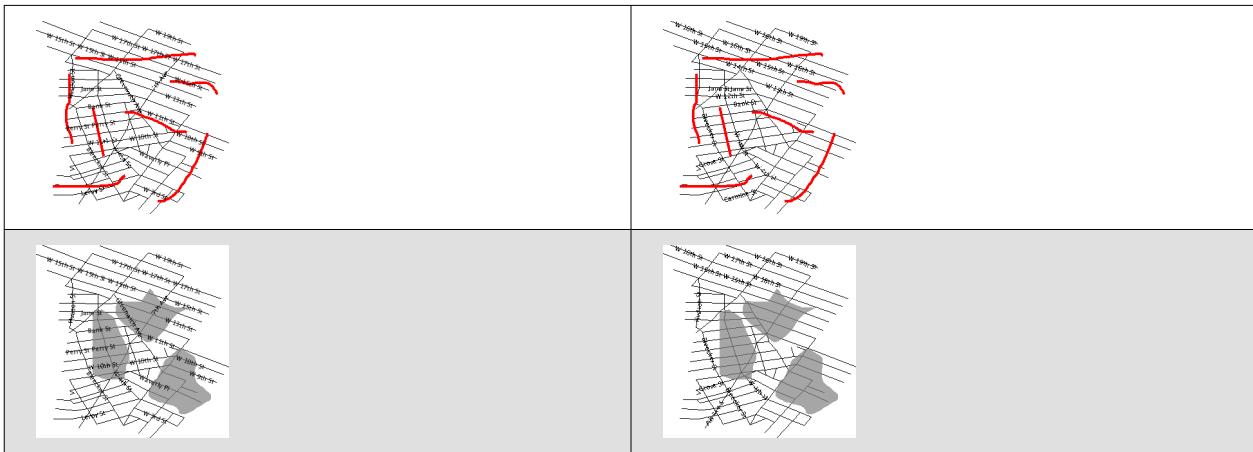


Applying the obstacle to a regular point style:

```
<PointSymbolizer>
  <Graphic>
    <ExternalGraphic>
      <OnlineResource
        xlink:type="simple"
        xlink:href="smileyface.png" />
      <Format>image/png</Format>
    </ExternalGraphic>
    <Size>32</Size>
  </Graphic>
  <VendorOption name="labelObstacle">true</VendorOption>
</PointSymbolizer>
```



Applying the obstacle to line/polygon style style:



### Avis

Beware of marking a line or poly symbolizer as a label obstacle. The label conflict resolving routine is based on the bounding box so marking as a label obstacle will result in no label overlapping not only the geometry itself, but its bounding box as

well.

## **SLD Tips and Tricks**

This section details various advanced strategies for working with SLD.

### **Dealing with mixed geometry types**

On occasion one might have the need to render data with a single geometry column whose content type can be different for each feature (some have a polygon, some have a point, etc).

SLD 1.0 does not provide a clean solution for dealing with such a case. This is due to a mix of two issues. The first one is that point, line, and polygon symbolizers can apply to other geometry types:

- Point symbolizers can apply to any kind of geometry; if the geometry is not a point, the centroid of the feature will be used in its place.
- Line symbolizers can apply to both lines and polygons.
- Polygon symbolizers can apply to lines as well, by adding a segment connecting the last point of the line to the first.

The second issue is that there is no standard way to apply a filter identifying the type of the chosen geometry attribute.

There are a number of workarounds, either requiring data restructuring or the use of non-standard filter functions.

### **Restructuring the data**

There are a few ways to restructure the data so that it can be rendered without difficulties using only standard SLD constructs.

#### **Split the table**

The first and obvious one is to split the table into a set of separate ones, each one containing a single geometry type. For example, if table findings has a geometry column that can contain point, lines, and polygons, three tables will be generated, each one containing a single geometry type.

#### **Separate geometry columns**

A second way is to use one table and separate geometry columns. So, if the table findings has a geom column, the restructured table will have point, line and polygon columns, each of them containing just one geometry type. After the restructuring, the symbolizers will refer to a specific geometry, for example:

```
<PolygonSymbolizer>
  <Geometry><ogc:PropertyName>polygon</ogc:PropertyName></Geometry>
</PolygonSymbolizer>
```

This way each symbolizer will match only the geometry types it is supposed to render, and skip over the rows that contain a null value.

#### **Add a geometry type column**

Une réponse efficace à la directive INSPIRE

A third way is to add a geometry type column allowing standard filtering constructs to be used, and then build a separate rule per geometry type. In the example above a new attribute, gtype will be added containing the values Point, Line and Polygon. The following SLD template can be used after the change:

```
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>gtype</ogc:PropertyName>
      <ogc:Literal>Point</ogc:PropertyName>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <PointSymbolizer>
    ...
  </PointSymbolizer>
</Rule>
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>gtype</ogc:PropertyName>
      <ogc:Literal>Line</ogc:PropertyName>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <LineSymbolizer>
    ...
  </LineSymbolizer>
</Rule>
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>gtype</ogc:PropertyName>
      <ogc:Literal>Polygon</ogc:PropertyName>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <PolygonSymbolizer>
    ...
  </PolygonSymbolizer>
</Rule>
```

All of the above suggestions do work under the assumption that restructuring the data is technically possible, which is usually true in spatial databases that provide functions that allow to recognize the geometry type.

### **Create views**

A less invasive way to get the same results without changing the structure of the table is to create views that have the required structure. This allows the original data to be kept intact, and the views to be used only for rendering sake.

### **Using non-standard SLD functions**

SLD 1.0 uses the OGC Filter 1.0 specification for filtering out the data to be rendered by each rule. A function is a black box taking a number of parameters as inputs, and returning a result. It can implement many functionalities, such as computing a trigonometric function, formatting dates, or determining the type of a geometry.

## Une réponse efficace à la directive INSPIRE

However, none of the standards define a set of well known functions. This means that any SLD document that uses functions is valid, although it is not portable to another GIS system. If this is not a problem, filtering by geometry type can be done using the geometryType filter function, which takes a geometry property and returns a string, which can (currently) be one of Point, LineString, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollection.

Using the function, a Rule matching only single points can be written as:

```
<Rule>
  <ogc:PropertyIsEqualTo>
    <ogc:Function name="geometryType">
      <ogc:PropertyName>geom</ogc:PropertyName>
    </ogc:Function>
    <ogc:Literal>Point</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <PointSymbolizer>
    ...
  </PointSymbolizer>
</Rule>
```

The filter becomes more complex if one has to match any kind of linear geometry. In this case, it would look like:

```
<Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:Function name="in3">
        <ogc:Function name="geometryType">
          <ogc:PropertyName>geom</ogc:PropertyName>
        </ogc:Function>
        <ogc:Literal>LineString</ogc:Literal>
        <ogc:Literal>LinearRing</ogc:Literal>
        <ogc:Literal>MultiLineString</ogc:Literal>
      </ogc:Function>
      <ogc:Literal>true</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <LineSymbolizer>
    ...
  </LineSymbolizer>
</Rule>
```

This filter would read like `geometryType(geom) in (LineString, LinearRing, MultiLineString)`. Filter functions in Filter 1.0 have a known number of arguments, so there are various in functions with different names, like `in2`, `in3`, ..., `in10`.

## Utilisateur

### Documentation utilisateur

Cette section rassemble la documentation autour du projet geOrchestra pour les utilisateurs de l'IDS.

### Visualiseur - mode d'emploi

# Une réponse efficace à la directive INSPIRE

**author:** Lydie Vinsonneau

## ***Introduction***

Pour appréhender les potentialités du visualiseur de geOrchestra, n'hésitez pas à cliquer sur la rubrique, parmi celles présentées qui vous intéresse :

Ergonomie générale de l'interface Barre d'outils zoom, mesure et impression Présentation des couches Ajouter une couche d'un serveur OGC distant Styler une couche (rubrique en cours de création) Requérir une couche (rubrique en cours de création)

## ***Ergonomie générale de l'interface***

La capture d'écran ci-dessous présente l'ergonomie de l'interface de visualisation de geOrchestra. Celle-ci laisse une grande part à la carte.

### **Légende :**

- Partie 1 : Outils de zoom, de déplacement, de mesure et d'impression ;
- Partie 2 : Changement d'échelle ;
- Partie 3 : Aide en ligne, légende et espace de travail d'enregistrement et de chargement des WMC ;
- Partie 4 : Travail sur les couches d'informations présentes ;
- Partie 5 : Géolocalisation grâce à Geonames, par référentiel ou par adresse ;
- Partie 6 : Mode connecté



## ***Barre d'outils zoom-mesure-impression***



En haut à gauche de la fenêtre, vous trouverez les outils suivants :

## Une réponse efficace à la directive INSPIRE

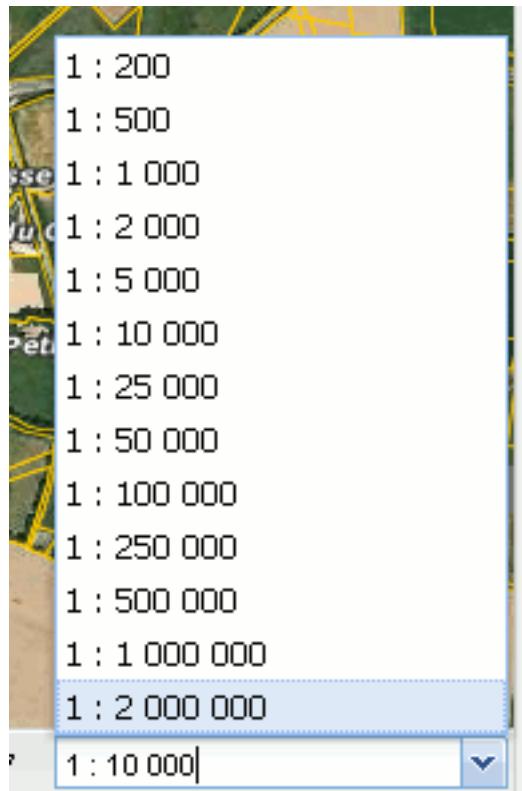
-  Zoom sur la **vue globale** ;
-  **Glisser/déplacer** sur la carte en conservant la même échelle ;
-  Zoom avant par clic souris ou rectangle ;
-  **Mesure d'une distance** ;
-  **Mesure d'une surface** ;
-  Accéder au **zoom précédent ou suivant** ;
-  **Imprimer** la carte.

### Changement d'échelle

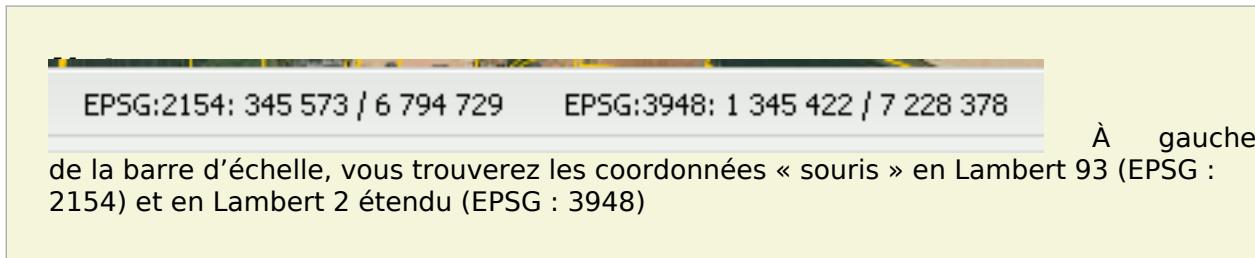
Les différentes échelles sont gérées par l'utilisateur dans le cadre bas à droite par l'utilisateur :



En déroulant le menu de changement d'échelle, vous pourrez choisir parmi les échelles définies pour le territoire :



Une réponse efficace à la directive INSPIRE



### Présentation des couches

Cocher/Décocher pour faire **apparaître/disparaître une couche d'information** :  Parcelle cadastrale ou  Parcelle cadastrale



Le bouton associé à une couche vous permet, après avoir cliqué dans la vue sur un objet de cette couche de récupérer en bas de la fenêtre les **informations descriptives liées à l'objet**.

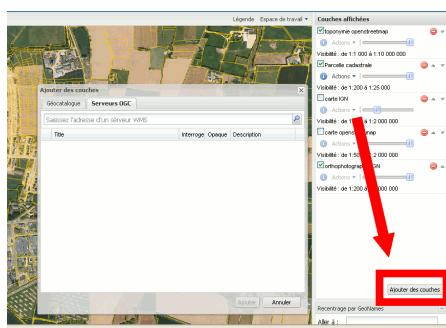
En déplaçant le curseur de la rampe associée à la couche, **vous pouvez jouer sur la transparence de celle-ci**.



Pour **monter une couche au premier plan ou descendre une couche en arrière-plan**, utilisez les flèches à droite du nom de la couche



### Ajout une couche d'un serveur OGC distant

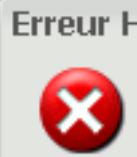


Une réponse efficace à la directive INSPIRE

Pour ajouter une couche d'un serveur OGC distant, cliquer sur le bouton Ajouter des couches

### Avis

Si le message suivant apparaît :



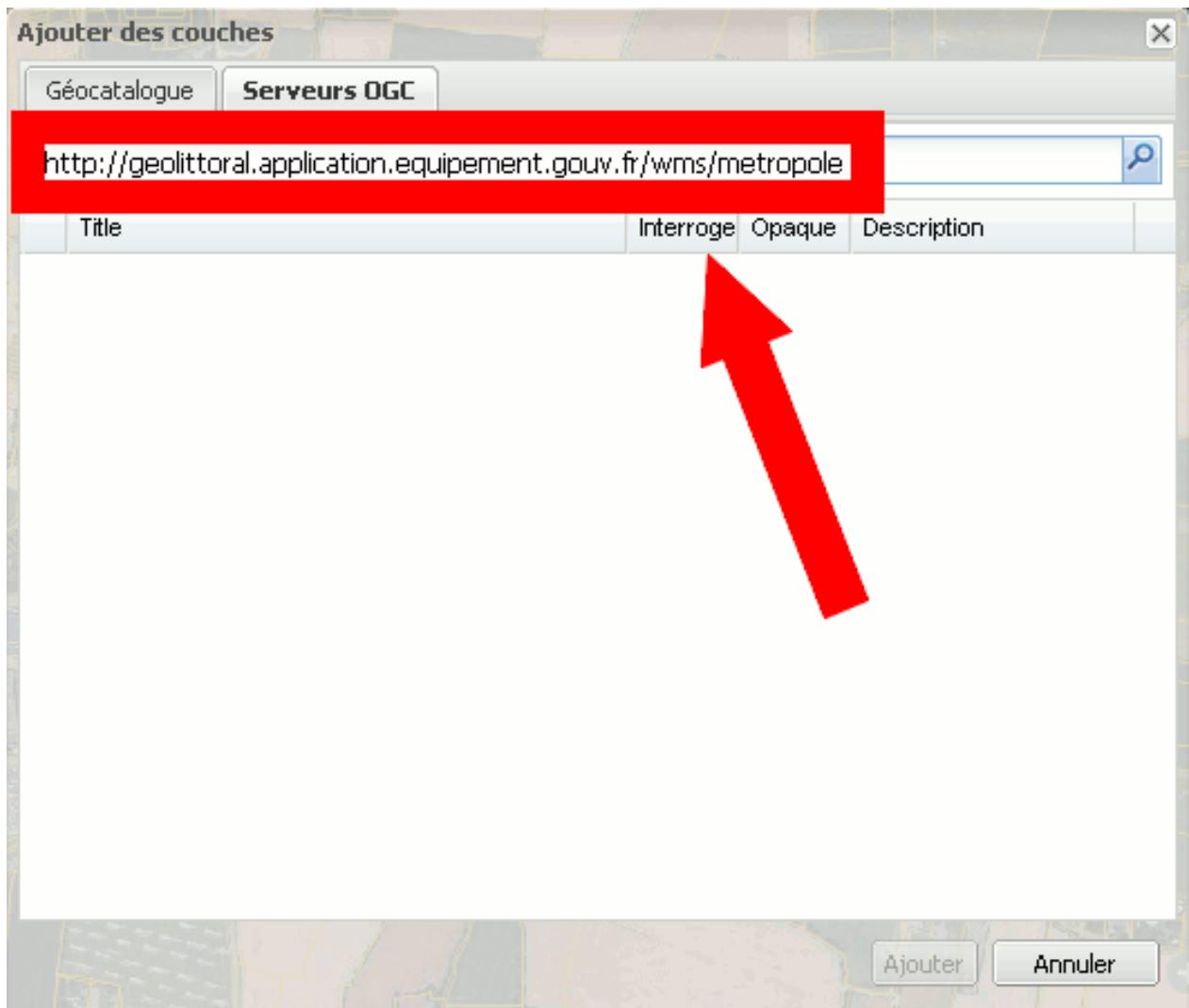
Une erreur est survenue.  
Pour plus d'information, nous vous invitons à chercher le code de retour sur [cette page](#).

OK

Cliquez sur OK pour continuer normalement

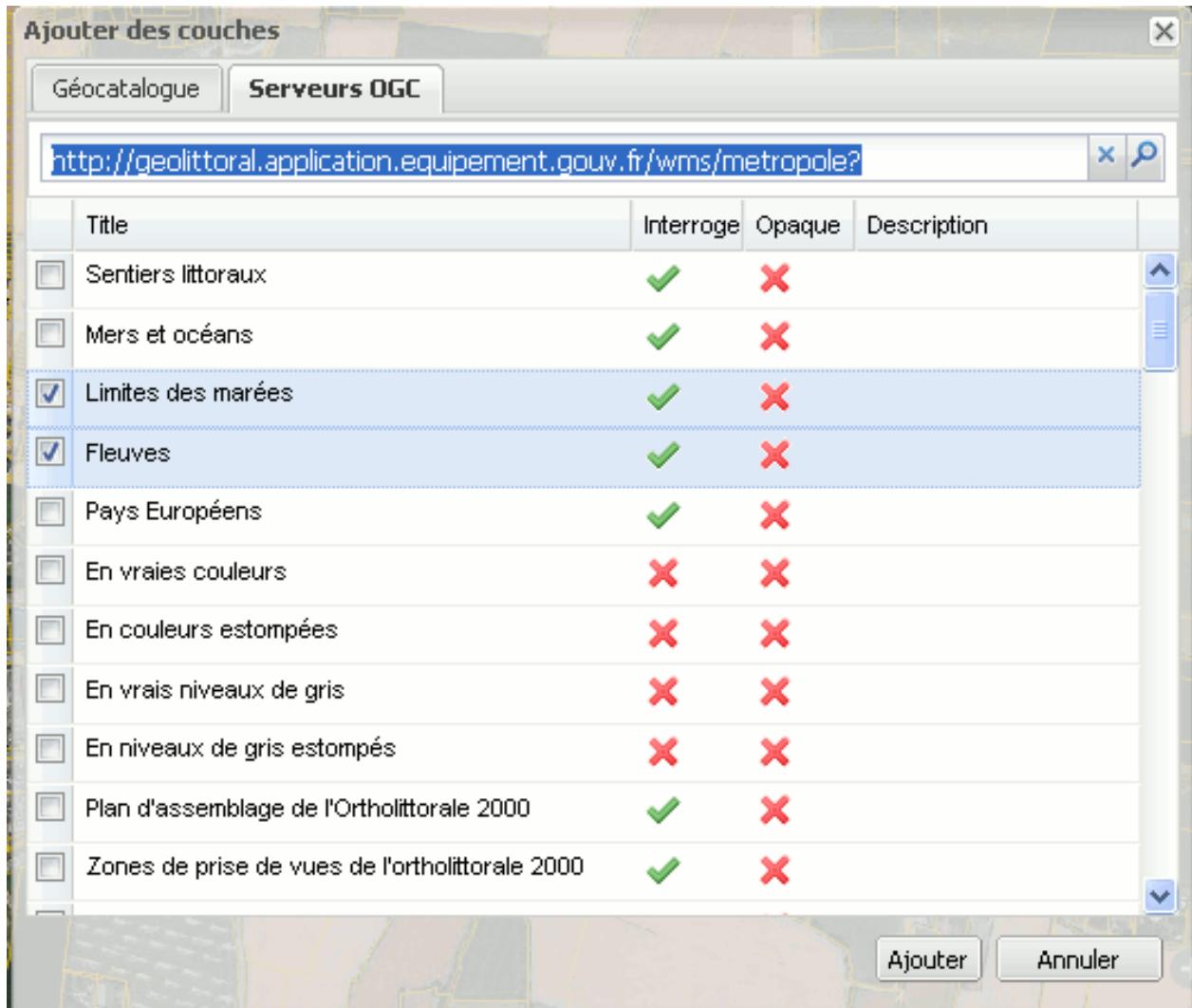
Cliquer sur l'onglet Serveurs OGC et renseigner l'URL du serveur OGC par exemple le serveur du Géolittoral : <http://geolittoral.application.equipement.gouv.fr/wms/metropole?> :

Une réponse efficace à la directive INSPIRE

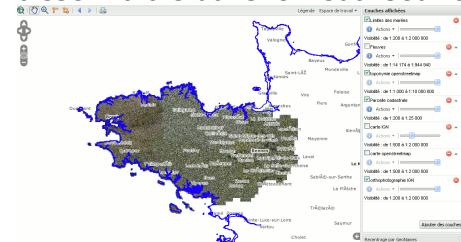


Vous pouvez ensuite **sélectionner les couches disponibles** qui vous intéressent de superposer dans le visualiseur aux autres couches :

Une réponse efficace à la directive INSPIRE

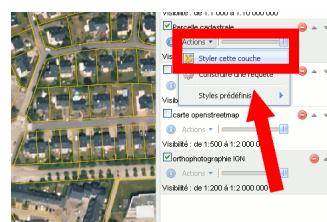


Cliquer sur Les couches apparaissent alors dans le visualiseur comme ci-dessous :



### Styler une couche

Nous allons montrer dans l'exemple ci-dessous, comment styler en rouge toutes les parcelles cadastrales < 500 m<sup>2</sup>



Une réponse efficace à la directive INSPIRE

Allez cliquer sur le bouton  **Actions ▾** de la couche « Parcelle cadastrale » Choisissez

### **Requérir une couche**

#### **Editeur - mode d'emploi**

**author:** Lydie Vinsonneau  
**author:** Yves Jacolin

#### **Introduction**

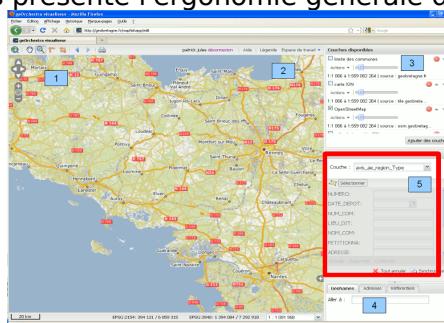
Le module d'édition de geOrchestra permet l'édition des couches spatiales diffusées. L'accès est lié aux droits que l'utilisateur possède sur les couches. Ainsi un utilisateur pourra éditer une couche ou non en fonction des droits au niveau du serveur.

Pour appréhender les potentialités de l'éditeur de geOrchestra, n'hésitez pas à cliquer sur la rubrique, parmi celles présentées qui vous intéresse :

- *Ergonomie générale de l'interface*
- *Charger une couche editable*
- *Modification d'un objet*
- *Saisie d'un objet*

#### **Ergonomie générale de l'interface**

La capture d'écran ci-dessous présente l'ergonomie générale de l'interface.



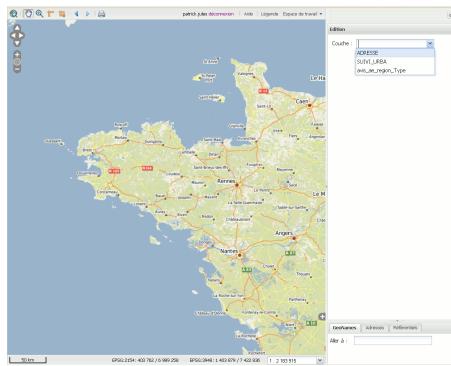
#### **Légende :**

- Zone 1 : Barre d'outils de zoom, de déplacement, de mesure et d'impression
- Zone 2 : Légende et espace de travail d'enregistrement et de chargement des WMC (Le standard WMC permet de sauvegarde l'état d'un projet dans un fichier pour s'en servir plus tard ou l'envoyer à un collègue).
- Zone 3 : Gestion des couches d'informations présentes (affichage, transparence, etc.);
- Zone 4 : Géolocalisation selon Geonames ;
- Zone 5 : Édition proprement dite.

#### **Charger une couche editable**

Pour charger une couche en mode édition, choisissez votre couche en haut à droite de la fenêtre :

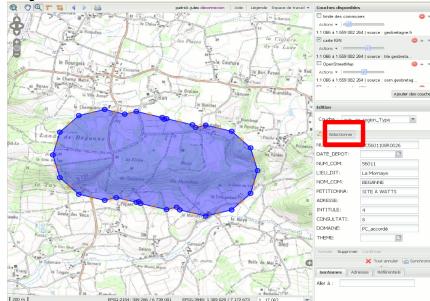
## Une réponse efficace à la directive INSPIRE



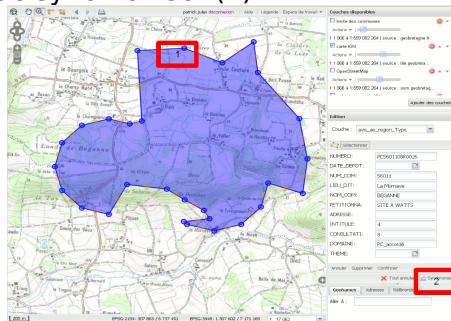
La couche est alors chargée en mode édition.

### **Modification d'un objet**

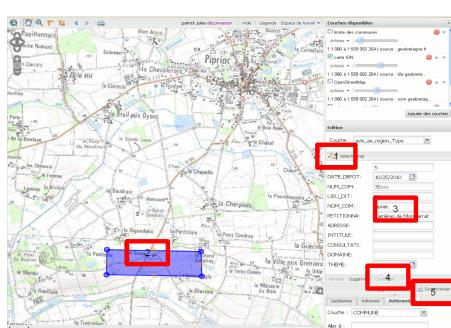
Le bouton *Sélection* associé à une couche vous permet, après avoir cliqué sur un objet de cette couche de modifier cet objet.



En déplaçant les points d'accroche du polygone vous pouvez modifier l'objet (1). Pour enregistrer, il est nécessaire de synchroniser (2).



### **Saisie d'un objet**



### **Légende :**

## Une réponse efficace à la directive INSPIRE

- Sélectionnez l'icone de saisie (1)
- Saisie du polygone(2)
- Saisie des valeurs(3)
- Confirmation/suppression(4)
- Synchronisation (5)

### Catalogue - mode d'emploi

**author:** Lydie Vinsonneau

#### Introduction

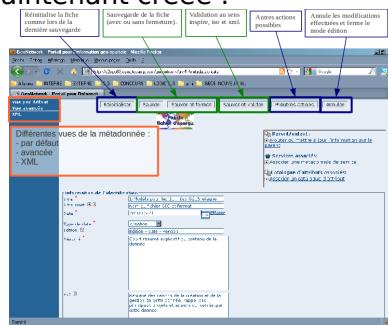
Pour appréhender les potentialités du catalogue de geOrchestra, n'hésitez pas à cliquer sur la rubrique, parmi celles présentées qui vous intéressent :

- *Création d'une fiche*
- *Saisie d'une fiche*
- *Import d'une fiche*
- *Administration d'une fiche*
- *Mise à jour d'une fiche de métadonnée et de la donnée*
- *Mise à jour d'une donnée issue d'une métadonnée de la V1*

#### Création d'une fiche

Après vous être connecté avec votre compte utilisateur, allez dans le menu administration → nouvelle métadonnée.

Puis, laissez par défaut le modèle "geOrchestra Inspire" et le groupe (votre structure) et cliquez sur créer. La fiche est maintenant créée :



#### Règles générales :

- En cliquant sur le symbol plus, vous rajoutez dans la fiche un champ identique à celui sur lequel vous cliquez.
- En cliquant sur la croix, vous supprimez le champ de la vue actuelle. Vous pouvez toujours le retrouver en allant dans la vue avancée associée à ce champ (ISO Complet pour tout voir).
- Ces boutons permettent de changer l'ordre des champs.

### Note

Si vous voulez ajouter un champ non présent dans le modèle, il vous suffira d'allez dans la vue avancée associée à votre champ (ISO Complet pour tout voir).

### Saisie d'une fiche

Nous allons ici détailler les différents champs à remplir (les champs notés avec une étoile rouge sont obligatoires).

#### Identification :

Information de l'identification	
Titre *	2-Modèle pour les données GéoBretagne
Titre court + ☒	-- Nom du fichier SIG et format --
Date *	2010-01-01T11:45:00 <input type="button" value="Effacer"/>
Type de date *	Création
Edition ☒	-- Edition - date - version --
Code *	-- FRA-SIREN-numéro d'ordre --
Résumé *	-- Court résumé explicatif du contenu de la donnée --
But ☒	-- Résumé des raisons de la création et de la gestion de cette donnée, rappel des principaux projets et acteurs concernés par cette donnée --
Etat + ☒	Mis à jour

- **Titre** : Il s'agit du titre littéral de la donnée. Il doit être clair et compréhensible par une personne n'ayant pas de connaissance particulière dans le domaine inhérent à cette donnée. Il devra comprendre la mention de l'entité géographique couverte. Ne pas mettre d'articles au début (le, la, les, ...).
- **Titre court** : titre court de la donnée.
- **Date** : C'est une date de référence concernant la donnée. On privilègera la date de validité à minima.
- **Code** : code de la forme : FR(code pays)-500-233-016(code siren du producteur)-nom technique (à valider).
- **Résumé** : Un résumé du contenu de la donnée. Il devra être accessible au grand public dans sa première partie et pourra être plus technique dans une seconde partie. Il pourra expliquer les champs « clés » de la donnée.

## Une réponse efficace à la directive INSPIRE

• **But** : Il décrit le contexte de la création de la donnée, les objectifs de la donnée et l'usage de la donnée qui a été fait après sa création.

• **Etat** : Etat de la donnée :

- Obsolète : donnée ancienne sans mise à jour prévue.
- Historique : donnée ancienne qui a été mise à jour depuis.
- Mis à jour : donnée en cours de création.
- Terminé (valeur par défaut) : donnée à jour (la mise à jour a été effectuée).
- En cours / requis / planifié : pas de définition exacte

### Point de contact :

Point de contact

Nom de la personne	Cellule SIG	Numéro de téléphone	
Organisation	Nom de la structure	Adresse	
Rôle *	Gestionnaire	Ville	
		Code postal	
		Pays	France
		France	
		Adresse e-mail	
Fréquence de mise à jour *		Lorsque nécessaire	
Remarque sur la mise à jour -- Commentaire éventuel sur cette fréquence			

Après clic sur la jumelle, accès à la liste des contacts préalablement saisis. Une fois la fenêtre ouverte, choisissez le contact puis cliquez sur Ajouter.

sharedObjectWindow

Nom	Email	Organisation	Rôle
YES!	email@address.com	Organisme	Position
New contact	email@address.com	Organisme	Position
Belo	email@address.com	Organisme	Position
ECAULT	loic.ecault@region-bretagne.fr	Région	technicien sig
Pôle Systèm	sig@region-bretagne.fr	Région Bretagne	Technicien SIG

• **Point de contact** : Il s'agit du point de contact si l'on veut des informations sur la donnée. Le nom de l'organisation, de la personne, le rôle de la personne et l'adresse e-mail sont obligatoires.

## Une réponse efficace à la directive INSPIRE

- **Rôle :** On privilègera le rôle Gestionnaire qui correspondra soit au service SIG de la structure ou au service interne de la structure qui gère la donnée.
- **Organisation :** Saisir le nom complet de la structure (et non les abréviations). Il devra commencer par une majuscule et ne pas comporter d'accents (ex : Communauté de communes de la Roche aux fees).
- **Fréquence de mise à jour :** fréquence de mise à jour de la donnée. Quelques précisions sur les valeurs non périodiques :
  - Continue : vrai dans le cas d'un capteur automatique qui met à jour la donnée en temps réel.
  - Lorsque nécessaire : quand on sait que la donnée sera mise à jour à une date non définie => valeur à privilégier par rapport à irrégulière ou inconnue.
- **Remarque sur la mise à jour :** Remarques éventuelles sur la fréquence de mise à jour. Par exemple l'événement qui déclanche la mise à jour ou la date de mise à jour.

### Mots clés :

The screenshot displays a user interface for managing descriptive keywords. It consists of two identical-looking sections, each with a header "Mots clés descriptifs". In the first section, the "Mot Clé" field contains "mots clés" and the "Type de mot clé" dropdown is set to "Thème". In the second section, the "Mot Clé" field contains "Bretagne" and the "Type de mot clé" dropdown is set to "Localisation". Both sections include a "+" button next to the "Mot Clé" field and a small icon above the "Type de mot clé" dropdown.

Lister les mots clés référents à la couche. Pour en rajouter, il faudra cliquer sur le symbol "plus" puis saisir les mots clés en minuscule. Les accents sont autorisés.

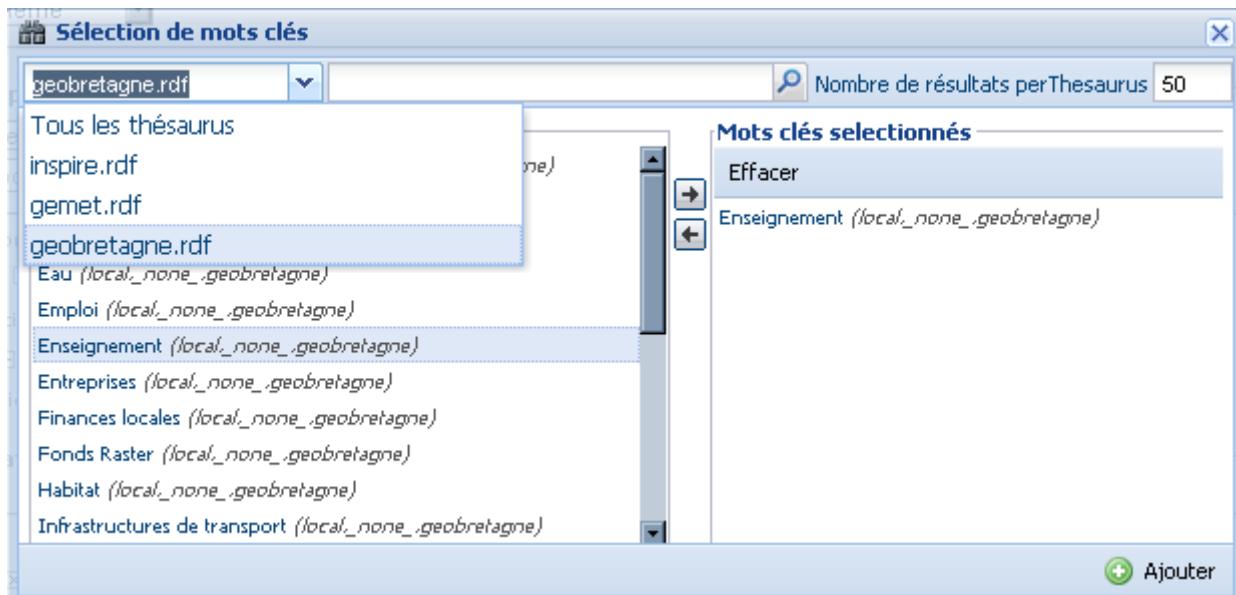
### Avis

Ne pas saisir les mots clés à la suite dans une même case. Utilisez le symbol "+" pour ajouter autant de case que nécessaire.

Pour accéder au thésaurus (symbole jumelle) voyez la section suivante.

### Sélection de mots-clés via un thésaurus :

## Une réponse efficace à la directive INSPIRE



- **Choix du thésaurus (liste déroulante):** il est fortement conseillé d'avoir au moins un mot-clé général et une autre GMET, par contre un seul mot-clé inspire.
- Pour sélectionner un mot-clé, double-cliquez sur le mot ou se servir des flèches.
- Pour finir, on clique sur Ajouter

### Conditions d'accès :

<input type="checkbox"/> Limitation d'utilisation    + <input checked="" type="checkbox"/> En dépôt des efforts et diligences mis en œuvre	
<input type="checkbox"/> Contraintes d'accès    + <input checked="" type="checkbox"/>	
<input type="checkbox"/> Contraintes d'utilisation    + <input checked="" type="checkbox"/>	
<input type="checkbox"/> Autres contraintes    + <input checked="" type="checkbox"/> Mentions obligatoires sur tout document de	
<ul style="list-style-type: none"> <li>• <b>Limitation d'utilisation :</b> par défaut, éventuelle contrainte d'utilisation.</li> <li>• <b>Contraintes d'accès :</b> Eventuelle contrainte d'accès à la donnée.</li> <li>• <b>Contraintes d'utilisation :</b> Eventuelle contrainte d'utilisation de la donnée.</li> <li>• <b>Autres contraintes :</b> Autres contraintes éventuelles comme la signification sur toute carte de la source de la donnée.</li> </ul>	
<input type="checkbox"/> Type de représentation spatiale    + <input checked="" type="checkbox"/> Vecteur	
<b>Résolution spatiale</b> + <input checked="" type="checkbox"/>	
<b>Résolution</b>	
<b>Echelle comparative</b>	
<b>Echelle</b>	
Dénominateur *	<input type="text" value="10000"/> (Suggestions : <input type="button" value="..."/>
Langue *	<input type="text" value="Français"/>
Jeu de caractère *	<input type="text" value="Utf8"/>
<b>Thématique</b> + <input checked="" type="checkbox"/>	
Thématique *	<input type="text" value="Limites politiques et administratives"/>

- **Type de représentation spatiale :** type de la donnée s'il s'agit d'un vecteur, raster, tableau...

## Une réponse efficace à la directive INSPIRE

- **Echelle** : Echelle de saisie de la donnée. (si inconnue, mettre l'échelle d'utilisation de la donnée).
- **Langue** : Langue de la donnée.
- **Jeu de caractère** : Type du jeu de caractère (par défaut UTF8).
- **Thématique** : Thème ISO à choisir dans la liste.

### Emprise :

Boîte géographique

WGS 84

Nord \*  
49.30250

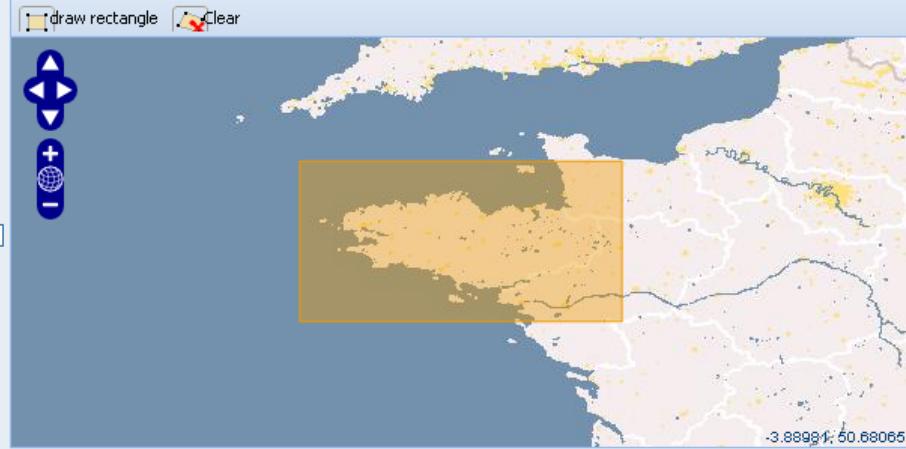
Ouest \*  
-5.43000

draw rectangle Clear

Sud \*  
46.96115

Est \*  
-0.72365

EPCL\_region  
COMMUNE  
DEPARTEMENT  
PAYS\_region



Pour saisir l'emprise de la donnée, plusieurs possibilités :

- Saisie « en dur » dans les zones de saisies (Nord, Sud, Est, Ouest).
- Saisie grâce aux couches référentielles :
  - Choix de la couche.
  - Saisie de l'entité sur laquelle portera l'emprise.
- Saisie grâce à la fonction draw rectangle

### Distributeur :

Distributeur

Nom de la personne *	-- Cellule SIG --
Organisation *	-- Nom de la structure --
Position	
Numéro de téléphone + *	
Adresse + *	
Ville *	
Code postal	
Pays *	France
Adresse e-mail + *	-- mail --
Rôle *	Distributeur
Nom du format *	shape
Version *	2010

### Informations diverses sur le distributeur de la donnée. Laissez le rôle

distributeur. **Nom du format** : format de distribution de la donnée. Il s'agira du format de la couche géographique vecteur (shape, table, mif/mid ...), raster (ecw, tif, ...) ou

## Une réponse efficace à la directive INSPIRE

autre. **Version** : version du format. Si la version est inconnue, mettre l'année de sortie du logiciel producteur de la donnée.

### Transfert de la donnée :

Attention, le fichier qui va être importé dans geobretagne doit être :

- Un fichier zip qui porte le même nom que le fichier shape qu'il contient.
- Le nom de la couche doit être de la forme donnee\_structure\_millesime (ex : route\_cg35\_2010).
- Le fichier .prj devra être au format geoserver. Par exemple, pour la projection Lambert 93, remplacer son contenu par les lignes suivantes :

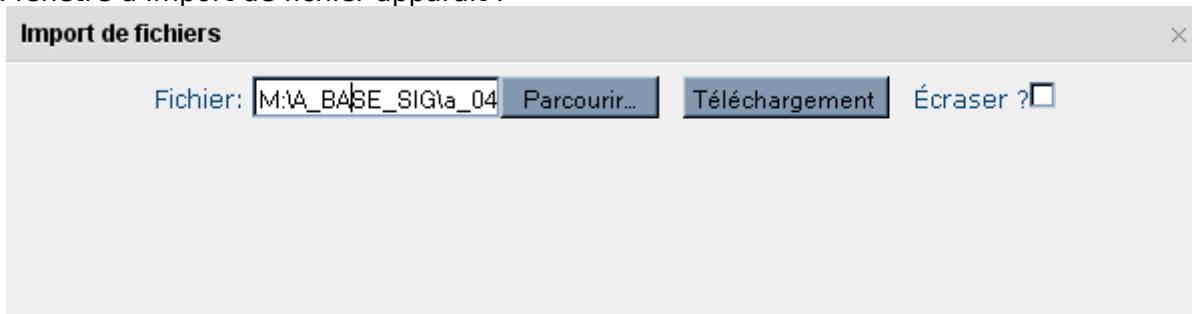
```
PROJCS["RGF93 / Lambert-93", GEOGCS["RGF93", DATUM["Reseau Geodesique Francais 1993", SPHEROID["GRS 1980", 6378137, 298.257222101], AUTHORITY["EPSG", "7019"]], TOWGS84[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], AUTHORITY["EPSG", "8901"], UNIT["degree", 0.017453292519943295], AXIS["Geodetic longitude", EAST, AUTHORITY["EPSG", "4171"]], PROJECTION["Lambert Conic Conformal (2SP)", AUTHORITY["EPSG", "9802"]], PARAMETER["latitude_of_origin", 46.5], PARAMETER["standard_parallel_1", 49.0], PARAMETER["false_easting", 6600000.0], PARAMETER["standard_parallel_2", 44.0], UNIT["m", 1.0], AXIS["Easting", EAST, AUTHORITY["EPSG", "2154"]]]
```

Ensuite, l'import de la donnée peut commencer :



Cliquez sur le bouton "Import de fichier" pour pouvoir importer un lot de données dans geonetwork

La fenêtre d'import de fichier apparaît :



Cliquez sur parcourir pour sélectionner le fichier zip (contenant votre donnée géographique) puis cliquez sur téléchargement. Un message s'affiche pour dire qu'il a été téléchargé avec succès, vous pouvez ensuite fermer la fenêtre.

### Publication OGC :

Cliquez sur geopublisher pour accéder au menu de publication OGC :

## Une réponse efficace à la directive INSPIRE

**Options de transfert** [x]

**Ressource en ligne**

URL libre *	<input type="text" value="http://install.georchestra.org:80/geonetwor"/>
Protocole	<input type="text" value="Fichier à télécharger"/>
Fichier	pays.zip
Description	[Texte]

**Supprimer** **GeoPublisher**

Puis réaliser les étapes suivantes :

- Choix du noeud (ici My preferred Geoserver node).
- Clic sur Vérifiez pour vérifiez que le jeu de donnée a été bien importée.
- Clic sur Publier pour faire la publication OGC de la couche.
- Clic sur Ajouter une source en ligne pour revenir à la fiche de métadonnées. Le lien de publication sera effectif.

**Options de transfert** [x]

**Ressource en ligne**

URL libre *	<input type="text" value="http://install.georchestra.org:80/geonetwor"/>
Protocole	<input type="text" value="Fichier à télécharger"/>
Fichier	pays.zip
Description	[Texte]

**Supprimer** **GeoPublisher**

### Note

S'il y a une erreur la première fois que vous cliquez sur publiez (ou vérifiez), ne pas hésitez à recommencer.

### Système de référence :

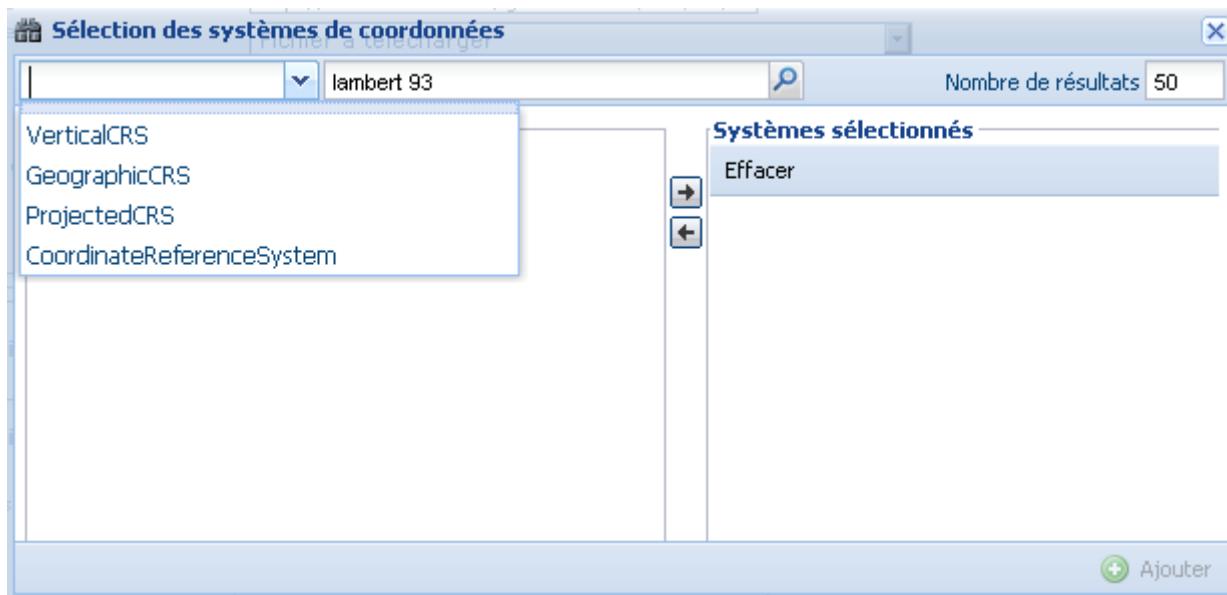
Cliquez ici pour accéder à la liste des projections :

**Information sur le système de référence** [x]

Code *	<input type="text" value="RGF93 / Lambert-93 (EPSG:2154)"/>
Nom de l'identifiant	<input type="text" value="EPSG"/>
Version	<input type="text" value="7.4"/>

Sélection par type de système de coordonnées ou saisie en direct de la projection choisie :

## Une réponse efficace à la directive INSPIRE



### Qualité de la donnée :

**Informations sur la qualité des données**

Niveau *	Jeu de données
Titre *	Titre de la donnée
Date *	2010-08-30 <input type="button" value="Effacer"/>
Type de date *	Création
Explication *	-- Degré de conformité de la ressource par rapport aux spécifications visées d'Inspire (Conforme, Non Conforme, Non Evalué) --
Réussi *	<input type="checkbox"/>
Généralités sur la provenance	-- Indique la couche géographique qui est à la source de cette donnée. Dans le cas où la donnée est issue d'une autre donnée source, elle est précisée. Ex: saisie depuis la base cadastrale --
Description de l'événement	-- Précise le référentiel de la donnée source --
Dénominateur *	25000 (Suggestions : <input type="button" value=""/>
Code *	RGF Lambert 93

- **Niveau** : Par défaut, jeu de données qui correspond à un ensemble de données de même type, dans une même couche ce qui sera la majorité des cas. Le type collection de données correspondra à une métadonnée qui correspond à plusieurs données (exemple de la BD\_Carto).
- **Titre** : Nom de la spécification relative à la qualité de la donnée. Par défaut, il s'agit de la spécification inspire.
- **Date/Type de date** : Date relative à la spécification (par défaut celle du 3/12/2008 relative aux fiches de métadonnées). On privilégiera le type de date, date de publication.

## Une réponse efficace à la directive INSPIRE

- **Explication :** Degré de conformité de la ressource par rapport aux spécifications visées d'Inspire (Conforme, Non Conforme, Non Evalué).
- **Réussi :** Cochez si la donnée est conforme à la norme INSPIRE.
- **Généralité sur la provenance :** Indique la couche géographique qui est à la source de cette donnée. Dans le cas où la donnée est issue d'une autre donnée source, elle est précisée. Ex: saisie depuis la base cadastrale.
- **Description de l'événement :** Référentiel utilisé pour saisir la donnée.
- **Dénominateur :** Echelle de référence pour la numérisation.
- **Code :** informations sur la projection de la donnée source.

### Autres informations :

<b>Métadonnées</b>			
Identifiant du fichier	2a2d0362-4d02-48de-a603-dbaa486cc32f		
Langue	<input type="button" value="Français"/>		
Jeu de caractère	Utf8: 8-bit variable size UCS Transfer Format, based on ISO/IEC 10646		
Date de création	2011-01-18T11:43:29		
Nom du standard de métadonnées	ISO 19115:2003/19139		
Version du standard de métadonnées	1.0		
<b>Contact</b>			
Nom de la personne	<input type="button" value="... Cellule SIG ..."/>	Numéro de téléphone	<input type="text"/>
Organisation	<input type="button" value="... Nom du partenaire ..."/>	Adresse	<input type="text"/>
Rôle *	<input type="button" value="Gestionnaire"/>	Ville	<input type="text"/>
		Code postal	<input type="text"/>
		Pays	<input type="button" value="France"/> France
		Adresse e-mail	<input type="text"/>

- **Langue :** langue de la fiche de métadonnée
- **Standard métadonnées ; version standard métadonnées :** A ne pas changer.
- **Contact :** Personne qui a saisi la fiche de métadonnée. On privilégiera le rôle de gestionnaire de la fiche de métadonnée. Pour l'organisation, elle devra commencer par une majuscule et ne pas comporter d'accents et de tirets. (ex : Communauté de communes de la Roche aux fees).

### Aide complémentaire :

- Lien INSPIRE by clouds .
- ` Manuel Geonetwork <[http://csm-bretagne.fr/redmine/attachments/598/Manual\\_fra.pdf](http://csm-bretagne.fr/redmine/attachments/598/Manual_fra.pdf)>` \_

### Import d'une fiche

Pour importer une fiche existante, aller dans le menu administration → import de fichiers XML.

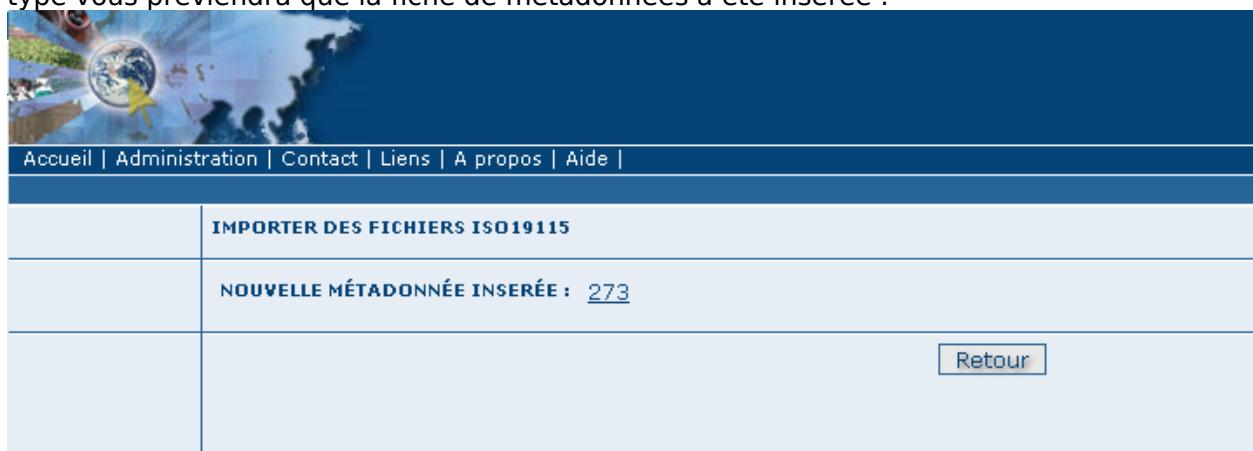
## Une réponse efficace à la directive INSPIRE

**IMPORTER DES FICHIERS ISO19115**

Mode d'insertion :	<input checked="" type="radio"/> Import de fichiers <input type="radio"/> Copier/Coller
Type de fichier :	<input checked="" type="radio"/> Fichier simple (XML, SLD, WMC...) <input type="radio"/> Fichier MEF
Métadonnées	<input type="text"/> Parcourir...
Type	<input type="button" value="Métadonnées"/>
Actions à l'import :	<input checked="" type="radio"/> Aucune action à l'import <input type="radio"/> Ecraser les métadonnées de même identifiant <input type="radio"/> Générer un nouvel identifiant pour les métadonnées insérées
Feuille de style	<input type="button" value="Aucun"/>
Valider	<input type="checkbox"/>
Groupe	<input type="button" value="editor_GR"/>
Catégorie	<input type="button" value="Aucun"/>

- **Mode d'insertion :** Choix entre import le fichier ou de copier/coller le contenu du fichier.
- **Type de fichier :** Format du fichier qui va être importé, le plus souvent XML. MEF étant un format d'export spécifique à geonetwork.
- **Métadonnées :** Chemin d'accès à la fiche de métadonnées à importer.
- **Type :** Métadonnées (pour une simple métadonnée) ou Modèle (pour une métadonnée qui servira de modèle aux futures saisies).
- **Action à l'import :** Actions à effectuer si la fiche est déjà présente dans la base.
- **Feuille de style :** Conversion d'une métadonnée issue d'une autre norme.
- **Valider :** Validation OGC et INSPIRE de la fiche de métadonnées.
- **Catégorie :** Type de la donnée issu de la fiche de métadonnées (le plus souvent Jeux de Données).

Une fois que vous avez saisi les paramètres voulu, cliquez sur insérer. Un message de ce type vous préviendra que la fiche de métadonnées a été insérée :



The screenshot shows a success message from the Geonetwork application. At the top, there's a decorative header featuring a globe and a map. Below it is a navigation bar with links: Accueil | Administration | Contact | Liens | A propos | Aide |. The main content area has a blue header bar with the text "IMPORTER DES FICHIERS ISO19115". Underneath, a message says "NOUVELLE MÉTADONNÉE INSÉRÉE : [273](#)". At the bottom right of the content area is a "Retour" button.

### Administration d'une fiche

Une fois la fiche créée (ou importée), nous avons accès à un certain nombre d'actions :

## Une réponse efficace à la directive INSPIRE

Résumé Lignes routières TER (Trains Express Régionaux) gérées par la région Bretagne.

Mots-clés car, ter, autocar, Bretagne, Réseaux de transport, Infrastructures de transport, transport en commun, transport de voyageurs, transport routier, localisation du transport, moyen de transport

Auteur: testadmin

Métadonnées Ajouter Editer Supprimer Autres actions Priviléges Catégories Crée un enfant Pas de

- **Ajouter** : duplique la fiche. On vous demandera de choisir le groupe où la fiche dupliquée sera affectée. Elle apparaîtra avec les mêmes informations que la fiche source.
- **Editer** : permet l'édition de la fiche de métadonnées.
- **Supprimer** : supprime la fiche de métadonnées.

Dans le menu *autres actions*, diverses actions sont possibles :

- priviléges ;
- catégorie ;
- créer enfant.

### Priviléges :

Lignes routières TER reviewer							
Groupes	Publication	Télécharger	Carte Interactive	Epingler	Editer	Notifier	
Tous	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			Tous Aucun
Intranet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Tous Aucun
Guest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Tous Aucun
editor_GR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tous Aucun
reviewer_GR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tous Aucun

Par défaut, seuls les membres de votre groupe ont les droits de visualisation (publication). Pour satisfaire aux exigences du partenariat Geobretagne, les cases **publication**, **télécharger**, **carte interactive** (lien avec le visualiseur) doivent être cochées pour **Tous**.

- **Epingler** : épingle la fiche en page d'accueil (favori).
- **Editer** : donner les droits d'édition (seulement possible sur un autre groupe, les membres du vôtre ont les droits d'édition par défaut).
- **Notifier** : Les membres du groupe seront avertis par mail si la fiche de métadonnées a été téléchargée.

### Catégorie :

Associer la fiche de métadonnée à la catégorie de donnée à laquelle elle fait référence.

Une réponse efficace à la directive INSPIRE



### Créer un enfant :

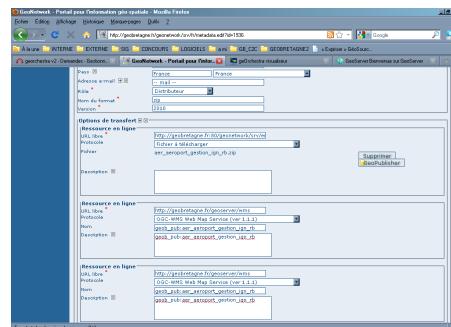
On vous demandera le groupe dans lequel vous voulez affecter la fiche enfant qui aura le même contenu que la fiche parent avec un lien en référence au parent.

### Mise à jour d'une fiche de métadonnée et de la donnée

Pour mettre à jour une fiche de métadonnée, il faut cliquez sur Editer dans l'encart de cette fiche. Le mode édition de la fiche apparaît à l'écran.



Pour mettre à jour la donnée, il faut allez dans l'encart Options de transfert puis cliquez sur Supprimer.



Une fois que l'on a fait cela, il suffit de reprendre la procédure de publication d'une donnée (voir page 6) et d'Enregistrer la fiche.

### Mise à jour d'une donnée issue d'une métadonnée de la V1

En effet, la structure au niveau du téléchargement et de la publication OGC n'est pas la même si la donnée est issue de la V1 ou si elle a été nouvellement créée.

A voir soit :

## Une réponse efficace à la directive INSPIRE

- fournir la donnée à un admin geobretagne
- voir si c'est possible de télécharger la donnée (après la suppression par un admin) puis de la télécharger. => avantage c'est qu'on devra avertir l'admin qu'une fois)

## F.A.Q.



## F.A.Q.

<b>Gouvernance</b>	<b>256</b>
À quoi sert le PSC (Project Steering Committee) ?	256
Comment contacter le PSC (Project Steering Committee) ?	256
Comment obtenir un accès à la forge ?	256
<b>Technique</b>	<b>257</b>
Quels sont les standards implémentés dans geOrchestra ?	257
Quelle est cette limite de 114 Ko qui apparaît parfois dans mapfishapp ?	257
La recherche par adresse ne fonctionne pas ...	257
Comment brancher la recherche référentiels ?	258
En mode édition, la page ralenti fortement, j'ai un message d'erreur	258
Comment étendre les types de format possibles dans l'extracteur ?	258
Impossible d'ajouter une couche raster (ecw, jpg2000, ...)	258

### **Gouvernance**

#### **À quoi sert le PSC (Project Steering Committee) ?**

Le PSC est le bureau du projet : il décide des grandes orientations du projet, permet de résoudre les problèmes (techniques ou non). Il est constitué de contributeurs majeurs de la communauté.

Ses membres sont élus par les "committers" du projet (voir le document [Comité de Pilotage du Projet \(PSC\)](#) pour plus d'informations).

#### **Comment contacter le PSC (Project Steering Committee) ?**

Une adresse permet de le contacter : psc chez georchestra point org.

#### **Comment obtenir un accès à la forge ?**

## Une réponse efficace à la directive INSPIRE

Le site de suivi des développements de GeOrchestra est maintenant ouvert à toute personne intéressée par l'actualité du projet et sa feuille de route.

Dans la phase actuelle de développement et jusqu'à la livraison de la version 1 stable, vous pouvez ainsi être informé en temps réel : annonces, soumissions de bogues, modifications du code source, recette, ...

Pour accéder à l'ensemble des ressources, trois étapes :

1. rendez-vous sur <http://applis-bretagne.fr/redmine/account/register> pour vous enregistrer sur la plate-forme avec l'identifiant et l'adresse de messagerie de votre choix 2. contactez les administrateurs du projet geOrchestra sur [info@georchestra.org](mailto:info@georchestra.org) en rappelant votre identifiant, afin qu'il valide votre inscription au projet. 3. une fois la confirmation reçue par messagerie, rendez-vous sur <http://applis-bretagne.fr/redmine/projects/show/geobretagne2> pour avoir accès à l'ensemble des informations

## **Technique**

### **Quels sont les standards implémentés dans geOrchestra ?**

geOrchestra utilise les standards proposés par l'OGC, notamment :

- [WMS](#) : Web Map Service
- [WMPS](#) : Web Map Tile Service
- [WFS](#) : Web Feature Service
- [WCS](#) : Web Coverage Service
- [WMC](#) : Web Map Context
- [SLD](#) : Styled Layer Descriptor
- [FE](#) : Filter Encoding
- [CSW](#) : Catalog Service

### **Quelle est cette limite de 114 Ko qui apparaît parfois dans mapfishapp ?**

C'est une limite dans le code javascript de la Mapfishapp, afin d'éviter des traitements potentiellement trop lourds côté client. On limite volontairement la taille des flux XML acceptable, et cette limite est évolutive selon le navigateur. Exemple : IE 6 => faible capacité à parser du XML => limite basse.

Il y a trois raisons pour que cela arrive :

- la liste de tous les codes EPSG est listée dans le GetCapabilities ;
- toutes les couches sont dans le même namespace ou vous n'utilisez pas de namespace dans l'URL des services (<http://monserver/geoserver/wms> par exemple) ;
- vous avez trop de couches dans le namespace (même raison que le point précédent).

Au final cela entraîne une taille trop importante pour le fichier XML du GetCapabilities.

Pour corriger ce problème :

- lister les codes EPSG qui doivent être disponibles dans GeoServer ;
- placer les couches dans différents namespaces ;
- utiliser l'url avec le namespace : <http://monserver/geoserver/monNamespace/wms>.

### **La recherche par adresse ne fonctionne pas ...**

## Communauté

Il faut installer le module [Préocédure d'installation du module search\\_address](#).

### **Comment brancher la recherche référentiels ?**

Voir [Procédure post-installation](#).

### **En mode édition, la page ralenti fortement, j'ai un message d'erreur**

Les performances JavaScript sont très dépendantes du navigateur web utilisé et IE n'est pas réputé pour avoir d'excellentes performances en la matière. Généralement on a ce problème dans l'éditeur lorsque l'on veut éditer une couche présentant une quantité élevée de sommets/points.

Il n'y a pas de solution.

### **Comment étendre les types de format possibles dans l'extracteur ?**

Ce n'est pas possible pour le moment. Vous pouvez financer cette évolution.

### **Impossible d'ajouter une couche raster (ecw, jpg2000, ...)**

Il faut installer des plugins supplémentaires dans GeoServer. Voir le blog de [geomatips](#) ou la documentation de GeoServer.

## Communauté

Le projet geOrchestra est un projet communautaire. Toutes les formes de contributions sont les bienvenues, que ce soit via la participation aux listes de discussion, la rédaction de documentation, les contributions au code (sous forme de patchs), ou encore par le financement de nouvelles fonctionnalités.

### **Listes de diffusion**

Le projet dispose actuellement de 4 listes :

- [georchestra@googlegroups.com](mailto:georchestra@googlegroups.com) est la liste généraliste sur geOrchestra. Elle concerne principalement les annonces sur le projet, l'entraide et les débats sur les évolutions fonctionnelles. L'inscription est libre.
- [georchestra-dev@googlegroups.com](mailto:georchestra-dev@googlegroups.com) est la liste consacrée aux développements autour de geOrchestra. Les discussions techniques pouvant naître sur la liste principale seront portées sur cette liste. Cette liste est également le lieu de construction et de vote des RFC ("Request For Comments", qui décrivent les façons de procéder pour améliorer le projet geOrchestra). L'inscription est libre.
- [georchestra-ci@googlegroups.com](mailto:georchestra-ci@googlegroups.com) est la liste consacrée à l'intégration continue des modules geOrchestra. L'inscription est libre.
- [georchestra-psc@googlegroups.com](mailto:georchestra-psc@googlegroups.com) est la liste utilisée par le PSC pour ses communications internes. L'inscription et la consultation des messages est réservée aux membres du PSC. Pour contacter le PSC, utilisez l'adresse psc chez georchestra point org.

### **Qui développe geOrchestra ?**

Liste des développeurs qui ont le droit de commit sur le SVN (par ordre alphabétique) :

- Bruno Binet (Camptocamp)
- Mathieu Coudert (Spot Image)

## Télécharger geOrchestra

- Jesse Eichard (Camptocamp)
- Pierre Giraud (Camptocamp)
- Philippe Huet (dotGee)
- Yves Jacolin (Camptocamp)
- Frédéric Junod (Camptocamp)
- Eric Lemoine (Camptocamp)
- Pierre Mauduit (eCairn)
- Fabrice Phung (GeoBretagne)
- François Van Der Biest (Camptocamp)

### ***Qui utilise geOrchestra ?***

- Région Bretagne, via le partenariat [GeoBretagne](#),
- Région Aquitaine, via le projet [PIGMA](#) mis en place par le GIP ATGeRI,
- Infrastructure de Données Spatiales Bolivienne via [GeoBolivia](#),
- GIP Bretagne Environnement,
- Projet [VIDAE](#) porté par l'UMR SAS INRA - Agrocampus Ouest à Rennes et l'UMR LISAH INRA - Supagro à Montpellier,
- INRA Champenoux,
- et bien d'autres ...

Contactez nous pour vous faire connaître !

## Télécharger geOrchestra

### ***Release***

Une version packagée complète de la suite sera bientôt disponible. Dans l'attente, nous vous recommandons d'installer à partir des sources (cf ci-dessous).

### ***Code source***

Le code source est mis à disposition sur le dépôt svn du projet :  
<http://svn.georchestra.org/georchestra/>

Pour télécharger les sources, vous devez installer *subversion* (*TortoiseSVN* sous Windows), puis :

```
svn co http://svn.georchestra.org/georchestra/ georchestra
```

### ***Packages***

Les archives au format WAR sont actuellement [téléchargeables](#) pour plusieurs modules : GeoServer, CAS, Doc, LDAP, ExtractorApp, et MapfishApp.

### ***Démonstration***

## À propos

Plusieurs instances publiques de geOrchestra permettent de répondre aux besoins de démonstrations.

Nous vous invitons à consulter l'[infrastructure de données spatiales de la région Bretagne](#) ou encore celle de la [région Aquitaine](#).

## À propos

### ***Partagez vos données avec vos partenaires***

geOrchestra permet de mettre en place une infrastructure de données spatiales afin de diffuser et partager des données entre partenaires ou vers un public plus large (mouvement openData, directive INSPIRE).

Ses différents modules permettent de gérer ces données : catalogue de données pour chercher et connaître les données diffusées, serveur cartographique pour diffuser les données sous forme de carte, extracteur pour récupérer les données dans un format "standard" permettant sa réutilisation au sein d'un SIG, éditeur pour modifier la donnée.

Ces différentes modules ont la capacité d'échanger des informations entre eux : visualiseur et catalogue pour proposer les couches disponibles ou retrouver les informations de métadonnées, visualiseur et serveur cartographique pour afficher les données sous forme de carte.

### ***Rejoignez la communauté***

Le projet geOrchestra est un projet communautaire, sa licence libre est un outil pour mutualiser les développements entre les différents contributeurs désirant s'impliquer dans le projet.

C'est aussi un moyen pour réunir une communauté de clients afin de mutualiser les demandes.