



deegree Web Map Service v2.4

lat/lon GmbH

Aennchenstr. 19
53177 Bonn
Germany
Tel ++49 - 228 - 184 96-0
Fax ++49 - 228 - 184 96-29
info@lat-lon.de
www.lat-lon.de

Dept. of Geography
Bonn University
Meckenheimer Allee 166
53115 Bonn

Tel. ++49 228 732098

Change log

Date	Description	Author
2006-11-10	Default behaviour for layer data sources adapted to changes in deegree implementation	Andreas Poth
2006-10-18	Description for accessing cascaded GetFeatureInfo request added	Andreas Poth
2006-11-03	Added details of supported SLD functionalities (Remote WFS servers) in section 4.1.4.	Markus Müller
2006-12-22	Added passed/added parameters for data sources.	Andreas Schmitz
2007-01-10	Update using new formatting style	Markus Müller
2007-05-22	Revision for demo version 2.1	Hanko Rubach
2007-12-11	Remove typos; reformatting	Judit Mays
2008-04-15	Update to v2.2	Judit Mays
2009-02-06	Include todos for version 2.3 of deegree	Andreas Poth
2009-09-01	Include pattern for external dataaccess datasource	Andreas Poth
2009-11-25	Updates for v2.3, mainly new data sources (incl. OSM) and vendor specific parameters.	Sebastian Goerke
2009-12-02	Updated figure 1 (WFS, WCS and WMS as data sources for deegree WMS)	Sebastian Goerke
2009-12-18	Updated path's (mainly localwcs and -wfs capabilities)	Sebastian Goerke
2009-12-21	Update for LOCALWFS and LOCALWCS description	Judit Mays
2010-01-10	Update documentation with new features implemented in deegree WMS	Andreas Poth
2010-02-25	New wms_configuration.xml (Appendix A); Reformatting	Sebastian Goerke

Table of Contents

1 Introduction.....	5
2 Download / Installation.....	7
2.1 Prerequisites.....	7
2.2 deegree Web Map Service release.....	7
2.3 Testing the installation.....	7
3 Architecture.....	10
4 Basic configuration.....	11
4.1 Structure of the configuration files.....	11
4.2 The deegree WMS configuration document.....	11
4.2.1 The DeegreeParam Element.....	12
4.2.2 Service Parameters.....	14
4.2.3 Capability/Request-Parameter.....	14
4.2.4 Exceptions / UserDefinedSymbolization.....	16
4.2.5 Layer.....	17
4.2.6 DataSource (within Layer).....	21
4.2.7 SRS / BoundingBox / Style (within Layer).....	29
4.3 Support for additional and vendor specific parameters.....	31
4.3.1 SQLTEMPLATES.....	31
4.3.2 WKT Parameter.....	31
4.3.3 filterproperty and filtervalue.....	31
4.3.4 TIME Parameter.....	32
4.4 Defining Styles.....	32
4.4.1 Label placement.....	36
5 Advanced configuration.....	39
5.1 Manual Tomcat integration.....	39
5.2 Offer own vector data (short description).....	42
Appendix A Example wms_configuration.xml.....	44
Appendix B Supported SRS.....	63
Appendix C Featuretype Definition.....	64
Appendix D Deployment Descriptor (web.xml).....	66

Index of Tables

Table 1: Directory structure of the WMS release.....	7
--	---

Illustration Index

Figure 1: Response of blue lake example GetMap request.....	8
Figure 2: Response of blue lake example GetMap request.....	9
Figure 3: WFS, WCS and WMS as data sources for deegree WMS.....	10
Figure 4: Files affecting the deegree WMS configuration.....	11

1 Introduction

deegree is a Java Framework offering the main building blocks for Spatial Data Infrastructures (SDIs). Its entire architecture is developed using standards of the Open Geospatial Consortium (OGC) and ISO Technical Committee 211 – Geographic information / Geoinformatics (ISO/TC 211). deegree encompasses OGC Web Services as well as clients. deegree is Free Software protected by the GNU Lesser General Public License (GNU LGPL) and is accessible at <http://www.deegree.org>.

deegree2 is the new release of deegree supporting a number of features deegree1 was not able to handle. This documentation describes setup and configuration of deegree Web Map Service (WMS), an implementation of OGC's Web Map Service Implementation Specification 1.1.1. deegree WMS is the official Reference Implementation of the OGC for the mentioned standard.

deegree's WMS is able to render vector data as well as raster data from different storage formats and deliver it to any client that is able to perform a HTTP GET or POST request. The currently supported storage formats are:

- PostgreSQL / PostGIS
- Oracle (Spatial / Locator)
- databases allowing JDBC-connections
- ESRI Shapefiles
- (all provided by the deegree WFS)
- GML2 and GML3 provided by a OGC WFS
- JPEG, GIF, PNG, BMP, TIFF and GeoTIFF images (provided by the deegree WCS)
- Images provided by any OGC compliant WCS
- Cascaded OGC WMS 1.0.0/1.1.0/1.1.1

Besides a WMS, deegree comprises a number of additional services and clients. A complete list of deegree components can be found at:

<http://www.lat-lon.de> → Products

Downloads of packaged deegree components can be found at:

<http://www.deegree.org> → Download

deegree's Web Map Service offers great flexibility regarding its configuration and adaption to different data sources, formats and server environments. The configuration of WMS is similar to the configuration of other deegree web

services and requires editing of different XML files which control the functionality of the server.

The web services of deegree are realized as Java modules controlled by one central servlet (a “dispatcher”). This servlet has to be integrated into the respective web server/servlet engine. Most of the common web servers support servlet technology, thus making deegree a universal product. The Apache-Tomcat 5.5 Servlet-Engine is recommended due to its widespread use and its status as an open-source product.

2 Download / Installation

2.1 Prerequisites

For deegree2 Web Map Service to run you need:

- Java (JRE or JSDK) version 1.5.x
- Tomcat 5.5.x

For installation of these components refer to the corresponding documentation at java.sun.com and tomcat.apache.org.

2.2 deegree Web Map Service release

deegree Web Map Service can be downloaded from <http://www.deegree.org>. The release is packed as a WAR-archive. Simply put this file into your `$TOMCAT_HOME$/webapps` directory and (re-)start Tomcat. The installation of deegree WMS is already done with this.

Note: It is also possible to extract the WAR archive into another place of your computer and direct Tomcat to this place. Because of this possibility, in the remainder of this document, the directory you extracted the files to is referred to as `wms_home` (`= $TOMCAT_HOME$/webapps/deegree-wms` in the standard case).

Your `wms_home` will contain the following structure:

directory	Content
<code>./WEB-INF</code>	Required by Tomcat, containing all libraries, configuration- and data-files
<code>./WEB-INF/conf/wms</code>	WMS configuration files
<code>./WEB-INF/data</code>	Example datasets
<code>./WEB-INF/scripts</code>	Scripts for creating and filling example databases

Table 1: Directory structure of the WMS release

2.3 Testing the installation

deegree WMS comes with three sample data sets:

- The 'blue lake' dataset that is used by OGC for testing WMS compliance
- the 'UTAH' dataset comprising spatial data of the state Utah (sources: Shape, Hsqldb, WCS)
- The 'OpenStreetMap' dataset including spatial data from the free wiki world map www.openstreetmap.org (sources: postgresSQL database, external data access class)

All of this data is automatically extracted and installed when unpacking the deegree WMS demo.

If installation succeeded, deegree WMS should be running by now with the sample data sets. Try the following request in your web browser:

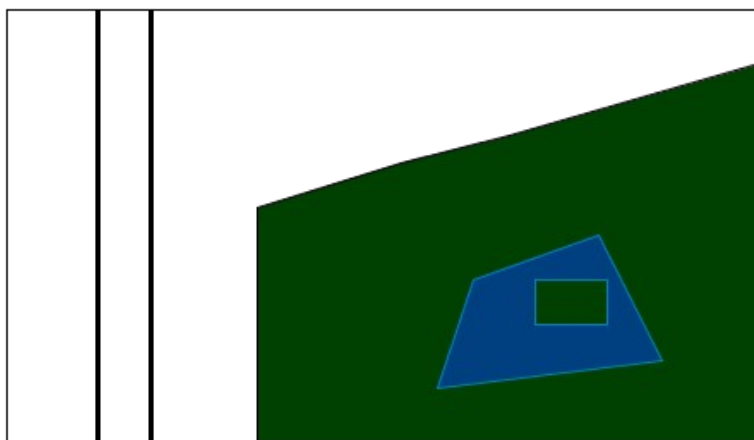
```
http://127.0.0.1:8080/deegree-wms/services?
SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities
```

This should provide a valid XML Capabilities file (if your browser asks for an application to open the file, try your browser again or any text editor). If this worked fine try a map request:

blue lake

```
http://127.0.0.1:8080/deegree-wms/services?
REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=467&HEIGHT=321&LAYERS
=Buildings,DividedRoutes,Forests,Lakes,MapNeatline&TRANSPARENT=TRUE&
FORMAT=image/png&BBOX=-0.004398,-
0.003327182608695652,0.004531852173913042,0.0028108956521739124&SRS=
EPSG:4326&STYLES
```

This request should deliver the following picture:

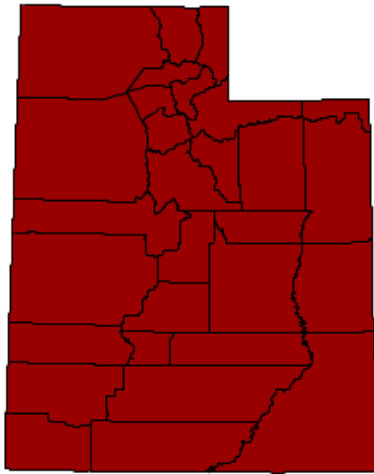


deegree 2.1 WMS 2007

Figure 1: Response of blue lake example GetMap request

UTAH

```
http://127.0.0.1:8080/deegree-wms/services?
REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=467&HEIGHT=321&LAYERS
=Counties&TRANSPARENT=TRUE&FORMAT=image/png&BBOX=210625,4175983,6978
11,4663170&SRS=EPSG:26912&STYLES
```

deegree 2.1 WMS 2007

Figure 2: Response of blue lake example GetMap request

If all of this worked fine, your deegree WMS is running and you can now add your own data. But let's first have a brief look at the architecture of deegree WMS.

3 Architecture

deegree2 WMS does not implement direct data access. All data that is used to create a map will be accessed via a WFS, WCS or another WMS. WFS and WCS can be realized locally – i.e. in the same Virtual Machine (VM) as the deegree WMS – or remotely as a web service (Figure 3). Different WMS layers can be delivered by different WFS, WCS, or WMS servers.

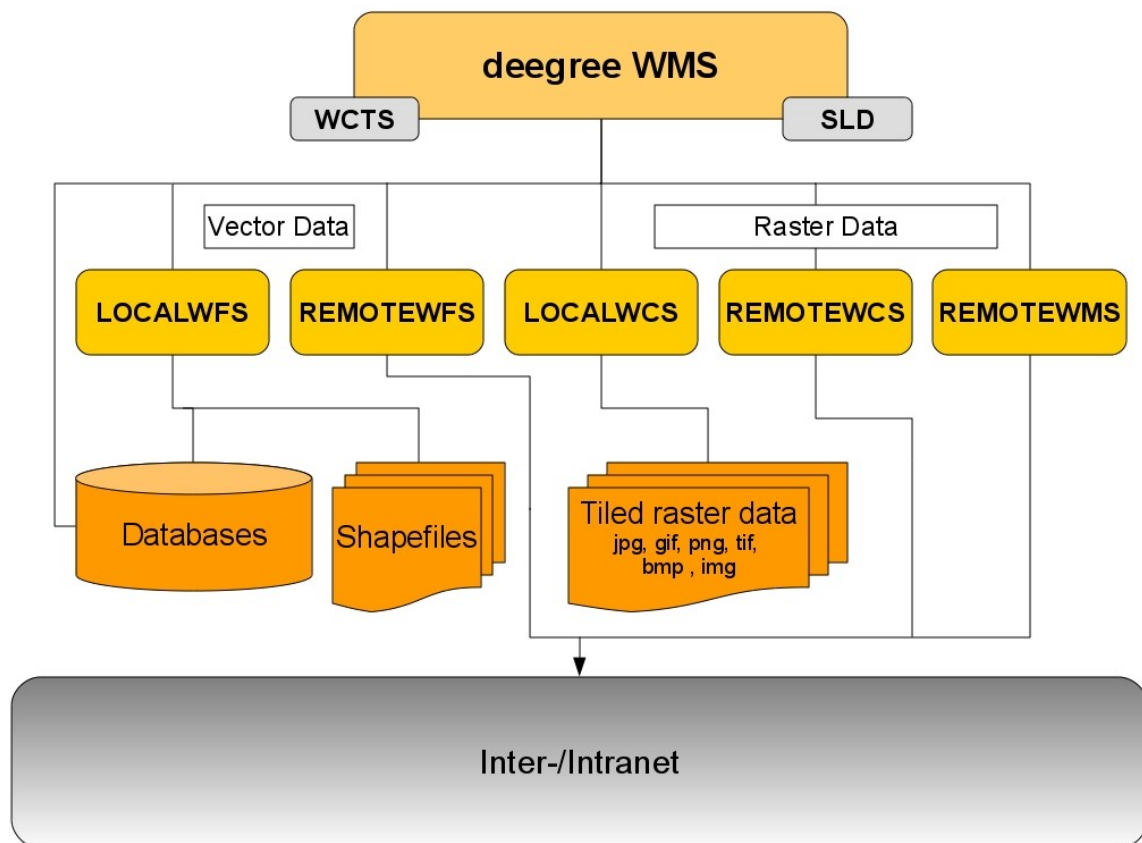


Figure 3: WFS, WCS and WMS as data sources for deegree WMS

For configuration purposes this means that if setting up a deegree WMS you also have to configure at least a LOCALWFS or LOCALWCS. Therefore it is recommended to consult these documentations as well if interested in advanced configuration options.

Since deegree WMS version 2.3, there is the possibility to use direct data access.

4 Basic configuration

4.1 Structure of the configuration files

The following figure shows the relationships between the different configuration files that have to be adapted:

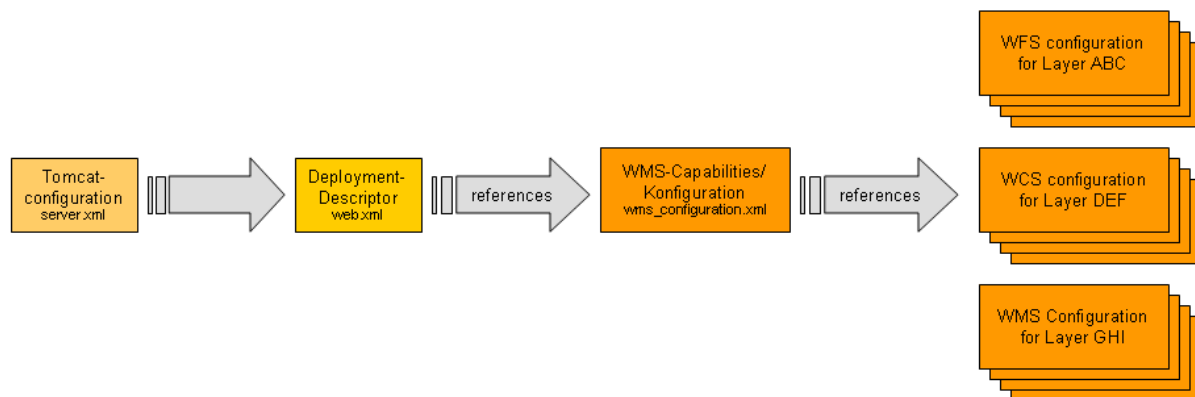


Figure 4: Files affecting the deegree WMS configuration

Besides the files shown in the diagram above, two additional files are part of the WMS configuration, the LOCALWFS_configuration.xml and the LOCALWCS_configuration.xml. (The default names are LOCALW*S_capabilities.xml) These two files are responsible for configuring the local WFS and WCS; please have a look at the deegree WFS- and WCS-documentation for details on how to do that.

4.2 The deegree WMS configuration document

The basic configuration allows use of the full set of parameters that can be used for controlling the deegree WMS using the deegree WMS configuration document. This includes the parameters mentioned in the OGC WMS 1.1.1 specification for the Capabilities document, but also additional specific technical parameters as for example descriptions of data sources for each layer or the maximum cache size. You can find the wms_configuration.xml file under \$wms-home\$/WEB-INF/conf/wms.

In the following, the elements of the configuration file will be described in detail. Appendix A includes the complete configuration document.

```

<?xml version="1.0" encoding="UTF-8"?>
<WMT_MS_Capabilities
  xmlns:deegree="http://www.deegree.org/wms"
  xmlns:sld="http://www.opengis.net/sld"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1.1"
  updateSequence="1.1.0">

```

In this code fragment, the namespace bindings are defined. It is usually preferable to just copy this fragment and use the prefixes defined here elsewhere in the document.

4.2.1 The DeegreeParam Element

After the root element, some deegree-WMS-specific parameters are defined. The `<DefaultOnlineResource>` is the URL by which the WMS operations can be invoked. This parameter can be overwritten by the URLs defined in the request-definitions. You have to adjust the `<DefaultOnlineResource>` to your system.

```
<deegree:DeegreeParam>
  <!-- The default online resource will be used for the following: If no
  LegendURL is provided, a GetLegendGraphic request with this URL will be
  deprecated automatically. If no OnlineResource element is specified within the
  Service element, this one will be inserted automatically. The operation
  specific online resources will NOT be filled automatically, since they also
  specify the HTTP method to be used. -->
  <deegree:DefaultOnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="simple" xlink:href="http://localhost:8080/deegree-wms/services"/>
  <!-- default = 100 (MB) -->
  <deegree:CacheSize>100</deegree:CacheSize>
  <!-- maximum lifetime of the processes in the WMS; default = 3600 (sec)-->
  <deegree:MaxLifeTime>3600</deegree:MaxLifeTime>
  <!-- maximum time for the execution of a request until an exception of time-
  exceed is thrown default 15 seconds -->
  <deegree:RequestTimeLimit>45</deegree:RequestTimeLimit>
  <!-- determines the quality of the map/image generated from a GetMap request.
  the parameter will only be evaluated if the desired format is able to handle
  it. The range of values is 0 ... 1, where 1 is best and 0 is worst. default is
  0.95 -->
  <deegree:MapQuality>0.95</deegree:MapQuality>
  <!-- maximum map width that can be requested. default = 1000 -->
  <deegree:MaxMapWidth>1000</deegree:MaxMapWidth>
  <!-- maximum map height that can be requested. default = 1000 -->
  <deegree:MaxMapHeight>1000</deegree:MaxMapHeight>
  <deegree:AntiAliased>true</deegree:AntiAliased>
  <!-- copyright note that will be drawn to the left bottom side of the maps ;
  you can also reference a graphic file using absolute path to file e.g.
  c:/images/mylogo.jpg -->
  <deegree:Copyright>deegree-WMS V.2.2 2008</deegree:Copyright>
  <!-- radius of the circle around the point a user has clicked to that will
  be considered for creating a search area for a GetFeatureInfo request;
  default = 5 -->
  <deegree:FeatureInfoRadius>10</deegree:FeatureInfoRadius>
  <!-- returns the URL where the DTD for OGC WMS capabilities DOCTYPE definition
  is located. default =
  http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd You can also
  reference this document locally if you run this service without internet
  connection -->
  <deegree:DTDLocation>
    <deegree:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="simple"
    xlink:href="http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd"/>
  </deegree:DTDLocation>
  <!-- define a proxy server if the WMS is separated by a firewall from the DTD-
  location or the locations of connected REMOTEWFS, REMOTEWCS and REMOTEWMS
  as an alternative you can set command line parameters at the call of the java
  interpreter like this: java -DproxyHost=131.220.106.104 -DproxyPort=1234
  -classpath ... -->
  <!--Proxy proxyHost="131.220.106.104" proxyPort="1234"/-->
</deegree:DeegreeParam>
```

The `<CacheSize>` parameter defines the size of the cache available to deegree-WMS in megabyte (this does not affect the cache for data sources). This parameter is optional, its default value is 100 MB. With the `<MaxLifeTime>` the user defines the time interval after which the internal processes of the WMS will be stopped if there are no active requests. If this parameter is not supplied, its default is 3600 seconds. By `<RequestTimeLimit>` the maximum time span is defined after which a request has to be processed. If this value is exceeded the processing is canceled and an exception will be thrown. Its default value is 15 seconds.

Afterwards a `<MapQuality>` parameter is defined, by which the quality of the created maps can be controlled. Values for this parameter are between 0 (lowest quality) and 1 (best quality) with the default value being 0.95. This parameter is only used for image formats supporting different quality values (e.g. jpeg).

The following two parameters limit the maximum size of a map that can be requested via WMS. It has to be considered that requesting large maps is very taxing on a server as the processing cost increases with the square of the border length of a map. For example, the creation a map having 1000x1000 pixel takes four times the memory than creating a map with size 500x500 pixel. Default is 1000 pixel for width and height.

The `AntiAliased` parameter defines whether deegree should use an anti aliasing filter while rendering the map. This usually improves the output for vector data. It is advisable to try both "true" and "false" with your data and see which looks better.

With the `<Copyright>` parameter it is possible to write a copyright mark in the lower left corner of each map. A copyright note can be either a text fragment or a reference to an image (e.g. `c:/images/copyright.gif`). In case this parameter is not supplied no output is produced.

The `FeatureInfoRadius` parameter defines a tolerance for the `GetFeatureInfo` request. If the user tries to click on a point and just misses it, he normally does not get back any data. Increasing this parameter to a value greater than 1 improves this behavior by expanding the search area around a point by the specified radius.

The `DTDLocation` parameter can be used to override the `DTDLocation` that is sent back in response to a `GetCapabilities` request. Since deegree nowadays does not try to load the DTD any more when encountering such a document, you'll usually drop this optional parameter, but if you wish to control the URL that is used in the `DOCTYPE` header of the Capabilities-document, here is where you set it.

If the WMS runs behind a firewall/proxy it is necessary to publish the proxy server address and port to the WMS (Java runtime environment) if data from remote data sources outside the firewall are included in the WMS. Because in this case

the WMS has to access servers from outside the firewall the network connection have to be passed through the proxy. Alternatively the proxy parameters can be passed to the Java runtime environment by command line parameters when starting the servlet engine (see java documentation).

4.2.2 Service Parameters

The following definition of service meta data is adopted from the WMS 1.1.1 specification.

```
<Service>
  <Name>WMS</Name>
  <Title>deegree2.2 demowms</Title>
  <!-- abstract and keywords are optional -->
  <Abstract>WMS reference implementation</Abstract>
  <KeywordList>
    <Keyword>deegree</Keyword>
    <Keyword>wms</Keyword>
  </KeywordList>
  <!-- The default online resource above will fill this one in for us, as well as
  in capabilities section. -->
  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
  xlink:href="http://localhost:8080/deegree-wms/services?" />
  <!-- the following service information is optional -->
  <ContactInformation>
    <ContactPersonPrimary>
      <ContactPerson>Andreas Poth</ContactPerson>
      <ContactOrganization>lat/lon</ContactOrganization>
    </ContactPersonPrimary>
    <ContactPosition>Technical Director</ContactPosition>
    <ContactAddress>
      <AddressType>Postal</AddressType>
      <Address>Aennchenstr. 19</Address>
      <City>Bonn</City>
      <StateOrProvince>NRW</StateOrProvince>
      <PostCode>53177</PostCode>
      <Country>Germany</Country>
    </ContactAddress>
    <ContactVoiceTelephone>0049228184960</ContactVoiceTelephone>
    <ContactFacsimileTelephone>00492281849629</ContactFacsimileTelephone>
    <ContactElectronicMailAddress>info@lat-lon.de
    </ContactElectronicMailAddress>
  </ContactInformation>
  <Fees>none</Fees>
  <AccessConstraints>none</AccessConstraints>
</Service>
```

4.2.3 Capability/Request-Parameter

The request-definition (GetCapabilities, GetMap and GetFeatureInfo) is in its form very similar to the definitions in OGC WMS 1.1.1 specification. However, some of the mandatory elements are declared optional. In case they are not supplied, deegree uses the corresponding default values. It is for example possible to not reference any of the requests explicitly. By default, GetCapabilities, GetMap and GetFeatureInfo will be initialized.

The following example is used to explain this by means of a GetMap request.

```
<GetMap>
  <!-- default = image/gif; image/png; image/jpg -->
```

```
<!-- mandatory = image/png -->
<Format>image/gif</Format>
<Format>image/png</Format>
<Format>image/png; mode 8bit</Format>
<Format>image/png; mode 24bit</Format>
<Format>image/jpeg</Format>
<Format>image/jpg</Format>
<!--Format>image/tif</Format-->
<!--Format>image/tiff</Format-->
<!--Format>image/bmp</Format-->
<DCPType>
  <HTTP>
    <Get>
      <!--
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple">
        <xlink:href="http://localhost:8080/deegree-wms/services?" />
      -->
    </Get>
    <Post>
      <!--
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="http://localhost:8080/deegree-wms/services?" />
      -->
    </Post>
  </HTTP>
</DCPType>
</GetMap>
```

You have to adjust the "xlink:href" attribute of the <OnlineResource> element to your system or leave the element commented out.

In case no <GetMap> element is defined, a GetMap with the formats image/gif, image/png and image/jpg will be initialized using the <DefaultOnlineResource> as web address for http-POST and GET. If a <GetMap> element is defined supplying information regarding Online-Resources but not the format, the above mentioned formats will be supported and initialized.

If only http-GET is defined, only this protocol is supported. Otherwise, if only http-POST is defined, http-GET is complemented as it is mandatory. The behaviour of the server regarding the image formats is likewise, all mandatory image formats will be initialized even if not defined by the user. An exception to this behaviour is the format image/png that is initialized anyway, as the OGC WMS 1.1.1 specification suggests the support of one format able to define an alpha-channel.

The behavior of the initialization for GetCapabilities is similar to GetMap. Configuration of the operations DescribeLayer, GetLegendGraphic, GetStyles and PutStyles is also optional, but they will not be filled in by default (if missing, the WMS will not support these operations).

The format definition of GetFeatureInfo-Requests is distinctly different to the other format definitions. Its default format is application/vnd.ogc.gml. However, further user-defined formats are possible. You can use XSLT scripts to support other formats. The XSLT scripts should take the standard GML3-Format as input and produce the desired format as output. You can specify the XSLT script by

annotating the <Format> element with a “filter” attribute as in the following (commented) example:

```
<GetFeatureInfo>
  <!-- default & mandatory = application/vnd.ogc.gml -->
  <Format>application/vnd.ogc.gml</Format>
  <Format>text/plain</Format>
  <Format>text/html</Format>
  <!-- <Format
  filter="file:///c:/deegreewms/wms/toXML.xsl">application/vnd.ogc.wms_xml</Format>
  -->
  <DCPType>
    <HTTP>
      <Get>
        <!-- <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple" xlink:href="http://localhost:8080/deegree-
        wms/services?" /> -->
        </Get>
      </HTTP>
    </DCPType>
  </GetFeatureInfo>
```

In this example the XSLT-script file:///c:/deegreewms/wms/toXML.xsl is used as filter for the output format 'application/vnd.ogc.wms_xml', but there is no filter definition for 'text/html'. In this special case deegree is using a default filter defined by \$wms-home\$/WEB-INF/conf/wms/featureinfo2html.xsl.

The <OnlineResource> element can be dropped, and will be filled in with the value of the “DefaultOnlineResource” as described above.

4.2.4 Exceptions / UserDefinedSymbolization

You can also define which exception formats are to be used in case a service exception occurs. The three possible values are shown below:

```
<Exception>
  <!-- default & mandatory= application/vnd.ogc.se_xml -->
  <Format>application/vnd.ogc.se_xml</Format>
  <Format>application/vnd.ogc.se_inimage</Format>
  <Format>application/vnd.ogc.se_blank</Format>
</Exception>
```

If this information is not supplied, the XML format will be used as default.

The next element defines whether the WMS is capable of supporting SLDs.

```
<UserDefinedSymbolization SupportSLD="1" UserLayer="1" UserStyle="1"
RemoteWFS="1"/>
```

A value of '1' means “true”, a value of '0' false.

This element as well as its attributes are optional. Its lacking indicates that the WMS will not support SLD.

SLD allows three alternatives to access remote servers (using GetMap Parameters, <remoteOWS> element or some default WFS); deegree WMS only

supports the second option: a <remoteOWS-element that is part of a <UserLayer>.

4.2.5 Layer

In the following, the layers known to the WMS are defined. Some elements here are not defined by the WMS 1.1.1 specification, especially those creating the association between layer and its data source.

It is possible, that a WMS supplies no layer definitions, indicating it serves no data of its own (a 'component' WMS as defined in the SLD specification). Such a server is nonetheless capable of serving maps by obtaining data directly or indirectly from the SLD documents sent to it.

The additional elements of the deegree-WMS configuration define the association of a layer with a data source (i.e. from a WFS or WCS). If a <Layer> has a <Name> but does not define a data source, using this layer with a GetMap or a GetFeatureInfo request causes that all <Layer>s nested within this layer will be affected by a request. For GetLegendGraphic requests only the top level layer will be used since GetLegendGraphic only allows the specification of one layer. If a REMOTEWMS (cascaded WMS) is used as datasource for a layer deegree is able to cascade GetLegendGraphic requests too if just one layer is requested from cascaded WMS within a datasource.

Simple reduced overview of example configuration:

```
<Layer>
  <Layer>nested Cite layers
    <Layer>Cite layer 1</Layer>
    <Layer>Cite layer 2</Layer>
    <Layer>Cite layer X</Layer>
  </Layer>
  <Layer>nested Utah layer
    <Layer>nested Utah administration
      <Layer>Utah county</Layer>
      <Layer>Utah Municipalities</Layer>
    </Layer>
    <Layer>Cite layer2</Layer>
    <Layer>Cite layerX</Layer>
  </Layer>
</Layer>
```

Full example for one nested layer:

```
...
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>citelayers</Name>
  <Title>Cite Layers</Title>
  <Abstract>deegree demo WMS</Abstract>
  <KeywordList>
    <Keyword>deegree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
  <MetadataURL type="ISO19115:2003">
    <Format>text/html</Format>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:type="simple" xlink:href="http://www.deegree.org" />
  </MetadataURL>
```

```

<!-- queryable="1" sets GetFeatureInfo to enabled. If set to "1" at least
one datasource must be set to "1" too -->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
<!-- please choose a unique <name> without spaces and a title which can
contain more details -->
  <Name>BasicPolygons</Name>
  <Title>BasicPolygons</Title>
  <Abstract>deegree demo WMS</Abstract>
  <KeywordList>
    <Keyword>deegree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
  <MetadataURL type="ISO19115:2003">
    <Format>text/html</Format>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:type="simple" xlink:href="http://www.deegree.org" />
  </MetadataURL>
  <MinScaleDenominator>0</MinScaleDenominator>
  <MaxScaleDenominator>10000000</MaxScaleDenominator>
  <!-- queryable="1" sets GetFeatureInfo to enabled for this datasource -->
  <deegree:DataSource failOnException="0" queryable="1">
    <deegree:Name>app:BasicPolygons</deegree:Name>
    <deegree:Type>LOCALWFS</deegree:Type>
    <deegree:OWSCapabilities>
      <deegree:OnlineResource xlink:type="simple"
        xlink:href="LOCALWFS_configuration.xml" />
    </deegree:OWSCapabilities>
    <!--name of the property that contains the geometries for this layer
    this element will only be resolved if the data source is a LOCALWFS or
    a REMOTEFWS -->
    <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
  </deegree:DataSource>
  <Style>
    <Name>default:BasicPolygons</Name>
    <Title>default:BasicPolygons</Title>
    <deegree:StyleResource>cite.xml</deegree:StyleResource>
  </Style>
  <!-- In case you do not wish to use the automatic legend generation you
  need to set the <sld:GetLegendGraphic> here. Refer to the documentation
  for details -->
</Layer>
<!-- The following Layer definition could be used as default for your own
layers -->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>Bridges</Name>
  <Title>Bridges</Title>
  <Abstract>deegree demo WMS</Abstract>
  <KeywordList>
    <Keyword>deegree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
  <MetadataURL type="ISO19115:2003">
    <Format>text/html</Format>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:type="simple" xlink:href="http://www.deegree.org" />
  </MetadataURL>
  <deegree:DataSource failOnException="0" queryable="1">
    <deegree:Name>app:Bridges</deegree:Name>
    <deegree:Type>LOCALWFS</deegree:Type>
    <deegree:OWSCapabilities>
      <deegree:OnlineResource xlink:type="simple"
        xlink:href="LOCALWFS_configuration.xml" />
    </deegree:OWSCapabilities>
    <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
  </deegree:DataSource>
  <Style>
    <Name>default:Bridges</Name>
    <Title>default:Bridges</Title>
    <deegree:StyleResource>cite.xml</deegree:StyleResource>
  </Style>
</Layer>

```

```

        </Style>
    </Layer>
</Layer>
...

```

The configuration given in the example above will enable to perform a GetMap or GetFeatureInfo request on layer 'citelayers'. Because 'citelayers' itself does not define its own data source but nests two layers its resulting map/info result will be the same as performing a GetMap or GetFeatureInfo request against 'BasicPolygons' **and** 'Bridges'.

The following configuration snippet shows how to define a cascading WMS layer, i.e., a layer which gets its data from a remote WMS layer. This is done by specifying a DataSource element as shown. Note that this layer also has **two** data sources. Requesting this layer will paint the two results one after another.

```

<Layer xmlns:app="http://www.deegree.org/app" queryable="1" cascaded="0" opaque="1"
noSubsets="1">
    <Name>Europe</Name>
    <Title>Datenlayer 1</Title>
    <Abstract>first testing layer</Abstract>
    <deegree:DataSource xmlns:deegree="http://www.deegree.org/wms"
        failOnException="1" queryable="1">
        <!-- default = equals the layer name and must be equal to the feature type
            or the layer name of the connected OWS -->
        <deegree:Name>app:Europe</deegree:Name>
        <!-- name of the property that contains the geometries for this layer .
            this element will only be resolved if the data source is a LOCALWFS or a
            REMOTEWFS, the value is case sensitive -->
        <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
        <!-- possible values: LOCALWFS, LOCALWCS, REMOTEWFS, REMOTEWCS,
            REMOTEWMS -->
        <!-- default = LOCALWFS -->
        <deegree:Type>LOCALWFS</deegree:Type>
        <!-- default value for OnlineResource@href is %Type%_capabilities.xml in
            the directory $wms-home$/WEB-INF/xml;
            default can only be used if data source type is LOCALXXX otherwise
            reference to the capabilities of the remote service must be set -->
        <deegree:OWSCapabilities>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                type="simple"
                xlink:href="http://www.lat-lon.de/documents/capabilities.xml"/>
        </deegree:OWSCapabilities>
        <deegree:ScaleHint min="0" max="1000"/>
        <!-- optional; the filter expression will be completed with the requested
            bounding box -->
        <deegree:FilterCondition>
            <wfs:Query typeName="app:Europe" xmlns:gml="http://www.opengis.net/gml"
                xmlns:ogc="http://www.opengis.net/ogc"
                xmlns:wfs="http://www.opengis.net/wfs">
                <wfs:PropertyName>app:Name</wfs:PropertyName>
                <wfs:PropertyName>app:Area</wfs:PropertyName>
                <wfs:PropertyName>app:Border</wfs:PropertyName>
                <ogc:Filter>
                    <ogc:And>
                        <ogc:PropertyIsLessThan>
                            <ogc:PropertyName>app:Area</ogc:PropertyName>
                            <ogc:Literal>50000</ogc:Literal>
                        </ogc:PropertyIsLessThan>
                        <ogc:PropertyIsLike wildCard="*" singleChar="?"
                            escape="\ ">

```

```

        <ogc:PropertyName>
            app:Administrator/app:City</ogc:PropertyName>
        <ogc:Literal>53115 Bonn</ogc:Literal>
    </ogc:PropertyIsLike>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</degree:FilterCondition>
</degree:DataSource>
<degree:DataSource failOnException="1" queryable="0">
    <!-- default = equals the layer name and must be equal to the feature type
        or the layer name of the connected OWS -->
    <degree:Name>app:EuropeanRivers</degree:Name>
    <!--name of the property that contains the geometries for this layer .
        this element will only be solved if the data source is a LOCALWFS or a
        REMOTEWFS
        default = 'app:GEOM' -->
    <degree:GeometryProperty>app:the_geom</degree:GeometryProperty>
    <!-- default is %Type%_capabilities.xml at the directory
        $wms-home$/WEB-INF/xml;
        default can only be used if data source type is LOCALXXX otherwise
        reference to the capabilities of the remote service must be set -->
    <degree:OWSCapabilities>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            type="simple"
            xlink:href="http://www.lat-lon.de/documents/another_capabilities_file.xml"/>
        </degree:OWSCapabilities>
    <degree:ScaleHint min="0" max="1000"/>
</degree:DataSource>
<degree:DataSource failOnException="0" queryable="0">
<degree:Name>TK500</degree:Name>
<degree:Type>REMOTEWMS</degree:Type>
<degree:OWSCapabilities>
    <degree:OnlineResource xlink:href="http://www.geoserver.nrw.de/GeoOgcWms1.3/
        servlet/NRW500?SERVICE=WMS&VERSION=1.1.0&REQUEST=GetCapabilities"
        xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"/>
</degree:OWSCapabilities>
<degree:FilterCondition>
    <degree:WMSRequest><![CDATA[Version=1.1.0&FORMAT=image/jpeg&
        TRANSPARENT=false&EXCEPTIONS=application/vnd.ogc.se_inimage&BGCOLOR=
        0xffffffff&LAYERS=Raster:UEK500:UEK500&STYLES=]]></degree:WMSRequest>
<degree:VendorspecificParameterDefinition>

    <degree:PassedVendorspecificParameter>
        <degree:Name>sessionid</degree:Name>
        <degree:Name>another</degree:Name>
    </degree:PassedVendorspecificParameter>

    <degree:AddedVendorspecificParameter>
        <degree:VendorspecificParameter>
            <degree:Name>user</degree:Name>
            <degree:Value>aUser</degree:Value>
        </degree:VendorspecificParameter>
        <degree:VendorspecificParameter>
            <degree:Name>password</degree:Name>
            <degree:Value>aPassword</degree:Value>
        </degree:VendorspecificParameter>
    </degree:AddedVendorspecificParameter>

</degree:VendorspecificParameterDefinition>
</degree:FilterCondition>
<degree:FeatureInfoTransformation>
    <degree:OnlineResource
        xlink:href="file:///C:/degree/wms/transformRemoteWMS.xsl"
        xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"/>
</degree:FeatureInfoTransformation>
<degree:ValidArea>
    <gml:Polygon srsName="EPSG:4326">
        <gml:outerBoundaryIs>

```

```

        <gml:LinearRing>
          <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </degree:ValidArea>
  <degree:TransparentColors>
    <degree:Color>#000000</degree:Color>
    <degree:Color>#FF0000</degree:Color>
  </degree:TransparentColors>
</degree:DataSource>
</Layer>

```

4.2.6 DataSource (within Layer)

After the optional element <Abstract>, zero, one or more <degree:DataSource> elements are defined. <degree:DataSource> can be provided multiple times in case a scale-dependent definition or an overlay is needed that uses different feature types and/or layers. A <degree:DataSource> may have two attributes:

- failOnException is of type boolean and defines if the processing of a GetMap request shall fail if accessing this data source fails. This is useful if a layer accesses more than one data source where at least one of these sources is not very stable (e.g. a REMOTEWMS) and one likes to avoid permanent failure of request processing. You can use '0' for "false" and '1' as "true", as above.
- queryable is also of type boolean and defines if a data source will be used for GetFeatureInfo requests. If the layer itself is queryable at least one of its data sources must be queryable.

The <degree:Name> element of the DataSource supplies the name of the FeatureType, Coverage or Layer of the data that is supplied by the afterwards described OWS. The <Name> must be defined.

After the FeatureType/Coverage/Layer name the properties of the data source are defined. If a WFS will be used as data source <degree:GeometryProperty> may be defined. As a WFS or a DATABASE can use arbitrarily defined properties to store geometries, this element is only needed if the data store is a LOCALWFS, DATABASE or a REMOTEWFS. Its default value is 'app:GEOM'.

The third subelement of <degree:DataSource> declares the type of the data source. Seven different values are possible for this:

- LOCALWFS → degree WFS running in the same Virtual Machine
- REMOTEWFS → any other WFS
- LOCALWCS → degree-WCS running in the same VM
- REMOTEWCS → any other WCS
- REMOTEWMS → other WMS which is used for cascading

- DATABASE → direct database access
- EXTERNALDATAACCESS → direct data access with a java class.

To be able to use the full set of opportunities the deegree WMS offers you should read the deegree WFS and WCS documentation. There will be explained how to configure access to a database and file-based raster and vector data sources. If you use a remote OWS as data source you may have to study its configuration documentation.

<deegree:Type> is also optional, its default value is LOCALWFS.

The optional element <deegree:ScaleHint> indicates the span of scales for which the respective data source is valid, it corresponds to the element defined by OGC WMS 1.1.1. If multiple data sources are defined for one layer, the spans supplied by <deegree:ScaleHint> must not overlap. If none is given, the ScaleHint of the layer itself is used (which may in turn inherit it from a parent layer).

The element <deegree:ValidArea> defines the area for which the data source is valid. Any polygon can be used for defining a valid area. This can be useful if a layer integrates more than one data source where each data source is responsible for a different area because it may be served by different vendors.

```
<deegree:ValidArea>
  <gml:Polygon srsName="EPSG:4326">
    <gml:outerBoundaryIs>
      <gml:LinearRing>
        <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
      </gml:LinearRing>
    </gml:outerBoundaryIs>
  </gml:Polygon>
</deegree:ValidArea>
```

Each datasource can have its own timeout. For this the Element <deegree:RequestTimeLimit> must be included into a <deegree:DataSource> Element. The time limit will be given in seconds; if no request time limit is defined 30 seconds will be the default.

4.2.6.1 Service-based datasources

<deegree:OWSCapabilities> specifies access to the capabilities of a remote or local data source. If for example a REMOTEWFS is defined as data source, the <deegree:OWSCapabilities> - respectively the <OnlineResource> - has to point to the Capabilities-document of the WFS.

If no <deegree:OWSCapabilities> is supplied, only the values LOCALWFS and LOCALWCS are valid, since the other types depend on the information supplied in this element.

In case neither <deegree:Type> nor <deegree:OWSCapabilities> are supplied it is assumed that the data source is a LOCALWFS whose Capabilities are in a document named 'LOCALWFS_capabilities.xml' in the directory \$wms-

home\$/WEB-INF/conf/wms/. If LOCALWCS is defined as <Type> and no <OWSCapabilities> element is supplied it is assumed that the Capabilities of the WCS are in a document named 'LOCALWCS_capabilities.xml' in the directory \$wms-home\$/WEB-INF/conf/wms/.

In case a WCS or a REMOTEWMS is defined as data source, it is necessary to assure that they can deliver their data as image/gif, image/png, image/tif, image/jpeg or as image/bmp, in order to ensure that deegree can actually read the image data.

```
<deegree:OWSCapabilities>
  <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
    xlink:href="http://www.lat-lon.de/documents/capabilities.xml"/>
</deegree:OWSCapabilities>
```

or

```
<deegree:OWSCapabilities>
  <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
    xlink:href="http://remote-wfs-server.de/servicename?SERVICE=WFS&
    VERSION=1.1.0&REQUEST=GetCapabilities"/>
</deegree:OWSCapabilities>
```

In case you use a local WFS but do not use the default capabilities (LOCALWFS_capabilities.xml) a file URL will be used to reference the capabilities document of the WFS.

```
<deegree:OWSCapabilities>
  <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
    xlink:href="file:///c:/documents/capabilities.xml"/>
</deegree:OWSCapabilities>
```

or, if the file is in the same folder as the wms_configuration.xml:

```
<deegree:OWSCapabilities>
  <deegree:OnlineResource xlink:type="simple"
    xlink:href="LOCALWFS_configuration.xml" />
</deegree:OWSCapabilities>
```

As next sub-element of <deegree:DataSource> a <deegree:FilterCondition> can be defined, that constrains the data delivered by a data source. For example a layer might be associated only with those features of the feature type 'app:Europe' (Type = REMOTEWFS or LOCALWFS), whose areas are smaller than 50000 m² and whose data administrator is working in Bonn. Please refer wms_configuration.xml to find a different example.

```
<!-- optional; the filter expression will be enhanced by the bounding box of the
current request -->
  <deegree:FilterCondition>
    <wfs:Query typeName="app:Europe" xmlns:gml="http://www.opengis.net/gml"
      xmlns:ogc="http://www.opengis.net/ogc"
      xmlns:wfs="http://www.opengis.net/wfs">
      <wfs:PropertyName>app:Name</wfs:PropertyName>
      <wfs:PropertyName>app:Area</wfs:PropertyName>
      <wfs:PropertyName>app:Border</wfs:PropertyName>
      <ogc:Filter>
        <ogc:And>
          <ogc:PropertyIsLessThan>
```



```

        <ogc:PropertyName>app:Area</ogc:PropertyName>
        <ogc:Literal>50000</ogc:Literal>
      </ogc:PropertyIsLessThan>
      <ogc:PropertyIsLike wildCard="*" singleChar="?"
escape="\ ">
        <ogc:PropertyName>
          app:Administrator/app:City</ogc:PropertyName>
        <ogc:Literal>53115 Bonn</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:And>
  </ogc:Filter>
</wfs:Query>
</deegree:FilterCondition>

```

The filter expression must be applicable to the defined type of the data source, meaning that for example the filter defined above can only be used in conjunction with a WFS, that stores features belonging to the feature type app:Europe. The typeName of the query in this case overwrites the value of the element <deegree:Name> defined above in the DataSource tag.

The following example shows the <FilterCondition> for a REMOTEWMS:

```

<FilterCondition>
  <WMSRequest><![CDATA[VERSION=1.1.1&LAYERS=Ortsteile, Bezirke&
    STYLES=style1,default&FORMAT=jpg&BGCOLOR=0xFFFFFF&
    TRANSPARENT=true]]></WMSRequest>
  <deegree:VendorspecificParameterDefinition>

    <deegree:PassedVendorspecificParameter>
      <deegree:Name>sessionid</deegree:Name>
      <deegree:Name>another</deegree:Name>
    </deegree:PassedVendorspecificParameter>

    <deegree:AddedVendorspecificParameter>
      <deegree:VendorspecificParameter>
        <deegree:Name>user</deegree:Name>
        <deegree:Value>aUser</deegree:Value>
      </deegree:VendorspecificParameter>
      <deegree:VendorspecificParameter>
        <deegree:Name>password</deegree:Name>
        <deegree:Value>aPassword</deegree:Value>
      </deegree:VendorspecificParameter>
    </deegree:AddedVendorspecificParameter>

  </deegree:VendorspecificParameterDefinition>
</FilterCondition>

```

In this case the filter expression is overwriting several parameters of the layers (Name of the data store, styles) and of an incoming request (VERSION, FORMAT, BGCOLOR, TRANSPARENT). Other parameters (SRS, WIDTH, HEIGHT, BBOX, EXCEPTION, REQUEST) must not be overwritten, as this might lead to inconsistencies when processing GetMap and GetFeatureInfo-Requests. If a REMOTEWMS does not support SRS used by a GetMap request against deegree WMS deegree will ensure that a REMOTEWMS will be requested with a SRS it supports and that the map returned by it will be transformed into desired SRS.

It is also possible to tell deegree which vendor specific parameters of an incoming request will be passed on to the remote server (the rest will be ignored) and which ones shall be automatically added to each request. This may be useful

if the remote WMS does not know about a session ID, but requires extra login information, for example.

In the above example, the parameters "sessionid" and "another" would be passed along with their values, as well as the additional "user" and "password" parameters with their values "aUser" and "aPassword".

To clarify, an "added" parameter is a parameter which will be added to each cascading request with its value. In the example, the remote WMS will receive GetMap requests like "...&request=GetMap&user=aUser&password=aPassword".

A "passed" parameter will cause the deegree WMS to make from a GetMap request like "...&request=GetMap&sessionid=abc123abc&another=andmore" a request to the remote WMS like "...TRANSPARENT=true&sessionid=abc123abc&another=andmore".

The VendorspecificParameterDefinition element is optional.

A filter for a WCS is similar to one for WMS:

```
<FilterCondition>
  <WCSRequest>
    <![CDATA[version=1.0.0&Coverage=world&Format=jpg]]>
  </WCSRequest>
</FilterCondition>
```

To all filters the bounding box of the GetMap request will be added dynamically for each request.

When assigning a RemoteWMS as a data source you may not just perform GetMap requests against it but also GetFeatureInfo requests. Because OGC WMS specification does not define a clear format for GetFeatureInfo responses deegree WMS may not be able to process results returned by a cascaded WMS. If a cascaded server returns an XML format you have the option to register an XSLT-script to the corresponding data source that transforms incoming XML documents into flat GML2 compliant feature collections. Registration of such an XSL script is done by adding the following element:

```
<deegree:FeatureInfoTransformation>
  <deegree:OnlineResource
    xlink:href="file:///C:/deegree/wms/sanitize.xsl"
    xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"/>
</deegree:FeatureInfoTransformation>
```

The XSLT script `sanitize.xsl` can be found in deegree SVN at `base/trunk/scripts/xslt/wms`; but you are free to use any other script that may do some special changes.

At last the element `<deegree:TransparentColors>` can be used with raster data sources (WMS, WCS). It defines a list of colours that shall be turned to be transparent. For each colour of a raster that shall be transformed to be transparent one `<deegree:Color>` element is added to

<degree:TransparentColors>. Colors are defined as hexadecimal RGB code with a leading '#'. The transparency will be a complete transparency with no opacity left.

```
<degree:TransparentColors>
  <degree:Color>#000000</degree:Color>
  <degree:Color>#FF0000</degree:Color>
</degree:TransparentColors>
```

4.2.6.2 Database-based data sources

Besides the service based data sources, there is the possibility to use a database as data source directly. In the <degree:Name> element, the name for the FeatureType is specified. The <degree:type> element declares the data source type as DATABASE. The following jdbc-namespace elements define the database which is used as data source. In the example, the data source is a postgresQL database. The associated entries can be found in the <jdbc:Driver> element and at the beginning of the url in the <jdbc:Url> entry. The <jdbc:Url> element specifies the database url. The <jdbc:User> and <jdbc:Password> elements contain the login information for the used database.

After defining the database connection the GeometryField of the used database relation must be named. Thereupon the database-based data source uses the SQLTemplate to determine the database relation which the layer should use. For that, it is important, that the SQL statement ends with where, like in the example. The NativeCRS element mentions the CRS used by the database relation. With the CustomSQLAllowed element it is possible to decide if custom SQL access is allowed. Possible values for that are "true" or "false". At least there is to mention, that it is necessary to change the style file. Without changing the degree name space there from app to DATABASE, styles will not work.

```
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>placeofworship</Name>
  <Title>PlacesOfWorshipBonn</Title>
  <Abstract>degree demo WMS</Abstract>
  <KeywordList>
    <Keyword>degree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
  <ScaleHint min="0" max="120" />
  <degree:DataSource failOnException="0"
    queryable="1">
    <degree:Name>app:placeofworship</degree:Name>
    <degree:Type>DATABASE</degree:Type>
    <jdbc:JDBCConnection xmlns:jdbc="http://www.degree.org/jdbc">
      <jdbc:Driver>org.postgresql.Driver</jdbc:Driver>
      <jdbc:Url>jdbc:postgresql://localhost:5432/osm</jdbc:Url>
      <jdbc:User>postgres</jdbc:User>
      <jdbc:Password></jdbc:Password>
    </jdbc:JDBCConnection>
    <degree:GeometryField>location</degree:GeometryField>
    <degree:SQLTemplate>select * from wms_bonn_religion where
    </degree:SQLTemplate>
    <degree:NativeCRS>EPSG:4326</degree:NativeCRS>
    <degree:CustomSQLAllowed>false</degree:CustomSQLAllowed>
  </degree:DataSource>
```

```

<Style>
  <Name>default:placeofworship</Name>
  <Title>PlacesOfWorshipBonn</Title>
  <degree:StyleResource>styles/osm_placeofworship.xml</degree:StyleResource>
</Style>
</Layer>

<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>MyData</Name>
  <Title>My database layer</Title>
  <ScaleHint min="0.001" max="1000000000" />
  <degree:DataSource failOnException="0" queryable="1">
    <degree:Name>MyData</degree:Name>
    <degree:Type>DATABASE</degree:Type>
    <jdbc:JDBCConnection xmlns:jdbc="http://www.deegree.org/jdbc">
      <jdbc:Driver>com.mysql.jdbc.Driver</jdbc:Driver>
      <jdbc:Url>jdbc:mysql://localhost:3306/db</jdbc:Url>
      <jdbc:User>admin</jdbc:User>
      <jdbc:Password></jdbc:Password>
    </jdbc:JDBCConnection>
    <degree:GeometryField>GEOM</degree:GeometryField>
    <degree:SQLTemplate>
      select LOCALITY_CODE, LOCALITY_NAME, value_h, AsBinary(GEOM) AS GEOM from
      locality where end_measure = "2008-12-14 00:00:00"
    </degree:SQLTemplate>
    <degree:NativeCRS>EPSG:4326</degree:NativeCRS>
    <degree:CustomSQLAllowed>true</degree:CustomSQLAllowed>
  </degree:DataSource>
  <Style>
    <Name>default:MyData</Name>
    <Title>Standard Legende für Mydata</Title>
    <degree:StyleResource>./styles/mydatastyles.xml</degree:StyleResource>
  </Style>
</Layer>

```

4.2.6.3 External Data Access

A third possibility for a data source is the external data access. This needs classes which specify the data source. Some classes are installed by default in deegree v.2.3. These ones are explained in this chapter.

4.2.6.3.1 OpenStreetMap Slippy Map Reader

The OpenStreetMap Slippy Map Reader is able to run tiles from the OpenStreetMap-Project (→ <http://www.openstreetmap.org>).

After defining a data source name with the <degree:name> element. The type of the data source is specified as EXTERNALDATAACCESS. The name of the class is org.deegree.ogcwebservices.wms.dataaccess.OSMSlippyMapReader. At least a Configurationfile is mentioned, in which the properties of this data source are stated.

```

<Layer queryable="0" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>OSMSlippyMapMN</Name>
  <Title>OSM Slippy Map Mapnik</Title>
  <Abstract>deegree demo WMS</Abstract>
  <KeywordList>
    <Keyword>deegree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
</Layer>

```

```

</KeywordList>
<ScaleHint min="0" max="1400" />
<degree:DataSource failOnException="0"
  queryable="1">
  <degree:Name>OSMMapnik</degree:Name>
  <degree:Type>EXTERNALDATAACCESS</degree:Type>
  <degree:ClassName>org.degree.ogcwebservices.wms.dataaccess.OSMSlippyMapReader</degree:ClassName>
  <degree:ConfigurationFile>dataaccess/osm/osmmapnik.properties</degree:ConfigurationFile>
</degree:DataSource>
</Layer>

```

4.2.6.3.2 WKT

A further possibility to make use of an external data source, is to use WKT. Using WKT makes it possible to add points, lines and polygons to a request and show them on the map.

The configuration of the layer is similar to the one of the OSM Slippy Map Reader. The class name and the ConfigurationFile differ from it as you can see in the example below.

```

<Layer fixedHeight="0" fixedWidth="0" noSubsets="0" queryable="0">
  <Name>position</Name>
  <Title>Position</Title>
  <Abstract>degree demo WMS</Abstract>
  <KeywordList>
    <Keyword>degree</Keyword>
    <Keyword>layer</Keyword>
  </KeywordList>
  <ScaleHint min="0" max="2800" />
  <degree:DataSource failOnException="0"
    queryable="1">
    <degree:Name>TestExternalDataAccess</degree:Name>
    <degree:Type>EXTERNALDATAACCESS</degree:Type>
    <degree:ClassName>
      org.degree.ogcwebservices.wms.dataaccess.WKTMarker
    </degree:ClassName>
    <degree:ConfigurationFile>
      ./dataaccess/wkt/wkt2.properties
    </degree:ConfigurationFile>
  </degree:DataSource>
  <Style>
    <Name>default:position</Name>
    <Title>default:position</Title>
    <Abstract>Position</Abstract>
    <degree:StyleResource>styles/wkt_position.xml</degree:StyleResource>
  </Style>
</Layer>

```

The properties file used for configuration looks like this:

```

# name of the feature type that will be created (important for style definition)
FeatureType={http://www.degree.org/app}:PositionType

# name of the vendor specific GetMap parameter that shall be evaluated
WKT=WKT2

```

The configuration given above makes a deegree WMS handling a vendorspecific parameter named WKT2. A valid WKT geometry must be assigned to this parameter as part of a GetMap request. deegree WMS will create a feature instance from the WKT with feature type 'PositionType' in namespace 'http://www.deegree.org/app'. For rendering engine of deegree there is no difference to features read from a WFS or a database, so style definition for a layer using WKT from vendorspecific parameters is the same as for any other layer based on vector data.

Example GetMap request:

```
http://$URL of a WMS$?
REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=300&HEIGHT=238&LAYERS
=Vegetation,position&TRANSPARENT=TRUE&FORMAT=image/png&BBOX=372233.9
342431239,4460438,489069,4552689&SRS=EPSG:26912&STYLES=&WKT2=LINSTR
ING(373000 4470000, 480000 4550000)
```

4.2.7 SRS / BoundingBox / Style (within Layer)

The following elements are optional elements defined by WMS 1.1.1. They are not modified by the deegree-configuration and not initialized with default values, save the SRS where a value of EPSG:4326 will be used if no other SRS is specified.

For bounding boxes care should be taken to use the correct SRS attribute, else clients won't be able to make use of it and the WMS may decide for (correct) requests to be outside and render the empty image instead of your data.

The LatLonBoundingBox should be specified at least once in the root layer, since the WMS specification says it's mandatory.

```
<!-- Just add the <SRS> you wish to support with your WMS -->
<!-- default = EPSG:4326 -->
<SRS>EPSG:4326</SRS>
<SRS>EPSG:26912</SRS>
<!-- It is strongly recommended that you set a LatLonBoundingBox; further
<BoundingBox>es are optional. Each layer inherits the BBoxes of the parent layer as
long as no explicit one is set. -->
<LatLonBoundingBox miny="-90" maxy="90" minx="-180" maxx="180" />
<!-- Setting <BoundingBox>es is optional but if set be sure to set the correct
SRS="EPSG:" -->
<BoundingBox SRS="EPSG:26912" miny="3581352" maxy="5432672" minx="0"
maxx="1504379" />
```

Note that the deegree WMS supports automatic coordinate reference system (CRS) for vector data (In WMS Specification the notion SRS instead of CRS is used). This means each layer based on a vector data set coming from a shape file or an Oracle Spatial database is available in nearly 300 CRS; you just have to define the ones you like for each layer or group of layers through <SRS> elements as in the code fragment above. A list of supported CRS is given in appendix B. There, you also find a java request to extract the latest list of supported CRS.

Next are the optional styles definitions for a layer. A `<degree:StyleResource>` element was added here, giving information about the resource where the style is defined. The resource where a style is defined must be a valid Styled Layer Descriptor document (see below)

```
<Style>
  <Name>teststyle</Name>
  <Title>ein teststyle</Title>
  <!-- here you set the name of the style and the location <degree:
StyleResource> where the WMS will find a style definition with the same
<UserStyle> name-->
  <degree:StyleResource>styles.xml</degree:StyleResource>
  <Abstract>dies ist nur ein test</Abstract>
  <LegendURL width="50" height="50">
    <Format>image/jpg</Format>
    <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
      xlink:href="http://www.degree.org/legend/teststyle.jpg"/>
  </LegendURL>
  <StyleSheetURL>
    <Format>/xsl</Format>
    <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
      xlink:href="http://www.degree.org/legend/teststyle.xsl"/>
  </StyleSheetURL>
  <StyleURL>
    <Format>/css</Format>
    <OnlineResource xlink="http://www.w3.org/1999/xlink" type="simple"
      xlink:href="http://www.degree.org/legend/teststyle.css"/>
  </StyleURL>
</Style>
```

Default value for `<degree:StyleResource>` is `'wms_home/WEB-INF/conf/wms/styles.xml'`.

Besides individually defined styles, each layer knows a default-style, that can be address either by the name 'default' or with a missing style parameter. If there is no style defined for a layer and cannot be inherited, such a layer knows exclusively the default style. degree-WMS allows definition of distinct default styles for each layer individually. For this purpose a style can be defined in `'wms_home/WEB-INF/conf/wms/styles.xml'` with its name beginning with 'default:' and displaying the name of the layer (default:%layername%). If for example a layer named 'Counties' is defined the name of its default style is default:Counties. If no individual default style is provided, the internal default style of degree is used (area = 50% gray, lines = 1 pixel width an black, points = 7 pixel square 50% gray).

4.3 Support for additional and vendor specific parameters

List of special parameters supported by degree WMS like TIME, SQLTEMPLATES, WKT etc. These vendor specific parameters are available since degree WMS 2.3.

4.3.1 SQLTEMPLATES

SQL TEMPLATES allow the use of SQL within requests. Therefore the data source of the corresponding layer has to be database-based. In the layer configuration, custom SQL access must be allowed.

4.3.2 WKT Parameter

With using WKT (well known text) it is possible to request user-defined points, lines and polygons (see above for detailed description).

4.3.3 filterproperty and filtervalue

Two vendorspecific parameters enables a client to request rendering just a defined list of features. Each feature to be rendered will be identified by the value of a given property. The name of the property is defined by the parameter filterproperty. The name of the property is not qualified so all properties with the given local name will be considered. A list of valid property values will be defined using parameter filtervalue. Each layer - or better its underlying data source - requested by a GeMap will evaluated for having a feature with a property with given name and one of the defined values. Just the features matching this filter condition will be rendered. Using the style addressed in a GetMap request.

Example:

```
http://$a deegree WMS URL$?
REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=381&HEIGHT=348&LAYERS=PV
%3ABesluitsubvlak&TRANSPARENT=TRUE&FORMAT=image/png&BBOX=104224.63009761559
,374443.4443868468,272322.2371852648,527981.4162149358&SRS=EPSG:28992&STYLE
S=&filterproperty=identificatie&filtervalue=NL.IMRO.9925.pv00003-0001
```

4.3.4 TIME Parameter

The Time Parameter can be used to present time-dependent data. The condition to use this parameter is, that the data source has got an ISO8601 conform attribute. In the example below the configuration for a layer which uses a time parameter is given. Two additional elements are necessary for the time parameter, the Dimension element and the DimensionProperty element inside the DataSource-block. A default time attribute is defined in the Dimension element. If the default time is not known by the data source, the WMS will paint nothing. A name and the used unit standard are also mentioned as attributes of the Dimension element there. If nearestValue is true, the wms gives out the nearest available time value for the requested time. MultipleValues allows the request of more than one time value at once.

Inside the DataSource element the DimensionProperty has to be specified. It has got a name and names the property which contains the time information.


```
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
  <Name>WeatherTime</Name>
  <Title>WeatherStationsWithTime</Title>
  <ScaleHint min="0.001" max="10000000" />
  <Dimension default="2009-01-01T00:00:00Z" name="time"
    units="ISO8601" multipleValues="true" nearestValue="true">
    2009-01-01T00:00:00Z/2009-01-07T00:00:00Z/PT5S</Dimension>
  <deegree:DataSource failOnException="0"
    queryable="1">
    <deegree:Name>app:WeatherTime</deegree:Name>
    <deegree:Type>LOCALWFS</deegree:Type>
    <deegree:OWSCapabilities>
      <deegree:OnlineResource xlink:type="simple"
        xlink:href="LOCALWFS_configuration.xml" />
    </deegree:OWSCapabilities>
    <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
    <deegree:DimensionProperty
name="time">app:timestamp</deegree:DimensionProperty>
  </deegree:DataSource>
  <Style>
    <Name>weathertime</Name>
    <Title>WeatherTime</Title>
    <deegree:StyleResource>styles/utah_weathertime.xml</deegree:StyleResource>
  </Style>
</Layer>
```

4.4 Defining Styles

Style definition for the deegree WMS will be done by creating one or more Styled Layer Descriptor (SLD) documents and assigning the UserStyles defined within it to one or more layers of the WMS. An SLD document that can be used for deegree WMS style definitions is valid against the OGC SLD 1.0.0 specification and defines one single 'NamedLayer'. The name of the <NamedLayer> element can be chosen freely and won't be evaluated currently. Within the <NamedLayer> element the user may define as many <UserStyle> elements as he likes. For better clarity we recommend to use more than one SLD document if the WMS offers a great amount of layers. The name of the 'UserStyles' corresponds to the name of the styles that can be assigned to a layer within the WMS configuration file as described above.

The following extract shows an example of an SLD

```
<?xml version="1.0" encoding="UTF-8"?>
<sld:StyledLayerDescriptor xmlns:sld="http://www.opengis.net/sld" xmlns:java="java"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xslutil="de.latlon.deejump.plugin.style.XSLUtility"
  xmlns:deegreewfs="http://www.deegree.org/wfs"
  xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:gml="http://www.opengis.net/gml"
  xmlns:app="http://www.deegree.org/app" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns="http://www.opengis.net/sld" version="1.0.0">
  <!-- Please be aware, that namespaces are used. So if you use featuretypes which
  are bound to app: the property names in this document must use them too. e.g.
  <ogc:PropertyName>app:geometry</ogc:PropertyName> -->
  <sld:NamedLayer>
    <sld:Name>Administration_etc</sld:Name>
    <sld:UserStyle>
      <!-- The <sld:Name> of the <sld:UserStyle> must be identical to the
      one referenced in the wms_configuration.xml -->
      <sld:Name>default:StateBoundary</sld:Name>
```



```

<sld:Title>default:StateBoundary</sld:Title>
<sld:IsDefault>0</sld:IsDefault>
<sld:FeatureTypeStyle>
  <sld:Name>SGID024_StateBoundary</sld:Name>
  <sld:Rule>
    <sld:Name>State Boundary</sld:Name>
    <sld:MinScaleDenominator>0</sld:MinScaleDenominator>
    <sld:MaxScaleDenominator>99999999</sld:MaxScaleDenominator>
    <sld:PolygonSymbolizer>
      <ogc:Geometry>
        <ogc:PropertyName>app:geometry</ogc:PropertyName>
      </ogc:Geometry>
      <sld:Fill>
        <sld:CssParameter name="fill">#00b400</sld:CssParameter>
        <sld:CssParameter name="fill-opacity">1.0</sld:CssParameter>
      </sld:Fill>
      <sld:Stroke>
        <sld:CssParameter name="stroke">#007d00</sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">1.0</sld:CssParameter>
        <sld:CssParameter name="stroke-width">3</sld:CssParameter>
        <sld:CssParameter name="stroke-dasharray">1</sld:CssParameter>
      </sld:Stroke>
    </sld:PolygonSymbolizer>
  </sld:Rule>
</sld:FeatureTypeStyle>
</sld:UserStyle>
<sld:UserStyle>
  <sld:Name>default:Municipalities</sld:Name>
  <sld:Title>default:Municipalities</sld:Title>
  <sld:IsDefault>0</sld:IsDefault>
  <sld:FeatureTypeStyle>
    <sld:Name>Municipalities 2004</sld:Name>
    <sld:Rule>
      <sld:Name>Municipalities</sld:Name>
      <sld:MinScaleDenominator>0</sld:MinScaleDenominator>
      <sld:MaxScaleDenominator>99999999</sld:MaxScaleDenominator>
      <sld:PolygonSymbolizer>
        <ogc:Geometry>
          <ogc:PropertyName>app:geometry</ogc:PropertyName>
        </ogc:Geometry>
        <sld:Fill>
          <sld:CssParameter name="fill">#e1b33f</sld:CssParameter>
          <sld:CssParameter name="fill-opacity">0.1</sld:CssParameter>
        </sld:Fill>
        <sld:Stroke>
          <sld:CssParameter name="stroke">#5f3d03</sld:CssParameter>
          <sld:CssParameter name="stroke-opacity">1.0</sld:CssParameter>
          <sld:CssParameter name="stroke-width">1</sld:CssParameter>
          <sld:CssParameter name="stroke-dasharray">1</sld:CssParameter>
        </sld:Stroke>
      </sld:PolygonSymbolizer>
    </sld:Rule>
    <sld:Rule>
      <sld:MinScaleDenominator>0</sld:MinScaleDenominator>
      <sld:MaxScaleDenominator>1600000</sld:MaxScaleDenominator>
      <sld:TextSymbolizer>
        <sld:Label>
          <ogc:PropertyName>app:name</ogc:PropertyName>
        </sld:Label>
        <sld:Font>
          <sld:CssParameter name="font-family">Arial</sld:CssParameter>
          <sld:CssParameter name="font-family">Sans-
            Serif</sld:CssParameter>
          <sld:CssParameter name="font-size">11</sld:CssParameter>
          <sld:CssParameter name="font-color">#000000</sld:CssParameter>
        </sld:Font>
        <sld:LabelPlacement>
          <sld:PointPlacement>
            <sld:Displacement>

```

```

        <sld:DisplacementX>0</sld:DisplacementX>
        <sld:DisplacementY>0</sld:DisplacementY>
      </sld:Displacement>
    </sld:PointPlacement>
  </sld:LabelPlacement>
</sld:TextSymbolizer>
</sld:Rule>
</sld:FeatureTypeStyle>
</sld:UserStyle>
</sld:NamedLayer>
</sld:StyledLayerDescriptor>

```

A special handling is used for RasterSymbolizer. Because SLD 1.0.0 is very according definition of raster styles deegree WMS uses Symbology Encoding 1.1.0 elements for definition. At the moment transparency, brightness, color ramps und classifications are supported.

The folloing SLD document shows a raster style classifying a noise raster (raster was available as 32 BIT Tiff without color map):

```

<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="String" xmlns="http://www.opengis.net/sld"
  xmlns:se="http://www.opengis.net/se" xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc" xmlns:app="http://www.deegree.org/app"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <NamedLayer>
    <Name>noise</Name>
    <UserStyle>
      <Name>default:noise</Name>
      <Title>RasterSymboliser</Title>
      <IsDefault>1</IsDefault>
      <FeatureTypeStyle>
        <Name>raster</Name>
        <Rule>
          <Name>raster2</Name>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>999999999</MaxScaleDenominator>
          <se:RasterSymbolizer>
            <se:Opacity>1</se:Opacity>
            <se:ColorMap>
              <se:Interpolate fallbackValue="#ddddd">
                <se:LookupValue>Rasterdata</se:LookupValue>
                <se:InterpolationPoint>
                  <se:Data>-9999</se:Data>
                  <se:Value>#FFFFFF</se:Value>
                </se:InterpolationPoint>
                <se:InterpolationPoint>
                  <se:Data>0</se:Data>
                  <se:Value>#33333333</se:Value>
                </se:InterpolationPoint>
                <se:InterpolationPoint>
                  <se:Data>10</se:Data>
                  <se:Value>#33FFFF</se:Value>
                </se:InterpolationPoint>
                <se:InterpolationPoint>
                  <se:Data>20</se:Data>
                  <se:Value>#FF33FF</se:Value>
                </se:InterpolationPoint>
                <se:InterpolationPoint>
                  <se:Data>30</se:Data>
                  <se:Value>#FFFF33</se:Value>
                </se:InterpolationPoint>
                <se:InterpolationPoint>
                  <se:Data>40</se:Data>
                  <se:Value>#3333FF</se:Value>
                </se:InterpolationPoint>
              </se:Interpolate>
            </se:ColorMap>
          </se:RasterSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>

```

```

        <se:InterpolationPoint>
          <se:Data>50</se:Data>
          <se:Value>#FF3333</se:Value>
        </se:InterpolationPoint>
        <se:InterpolationPoint>
          <se:Data>99999</se:Data>
          <se:Value>#FFFF0</se:Value>
        </se:InterpolationPoint>
      </se:Interpolate>
    </se:ColorMap>
  </se:RasterSymbolizer>
</Rule>
</FeatureTypeStyle>
</UserStyle>
</NamedLayer>
</StyledLayerDescriptor>

```

For a detailed description of defining a Styled Layer Descriptor document please have a look at the OGC SLD 1.0.0 specification. deegree supports most forms of style definition.

One additional feature of the deegree WMS in context of defining styles that has been mentioned before is its capability of using an individual default style for each layer. To realize this for each layer a <UserStyle> must be defined that has a name starting with 'default:' followed by the layer's name (e.g. default:StateBoundary is the default style for the layer StateBoundary; see SLD fragment above). So if the default style is defined in a GetMap request the individual default style will be selected. If you don't define an individual default style 50% gray for fills and 1 pixel thick black lines will be used.

TIP:

To convert existing ESRI .avl visualization rules to SLD use the following command

```
java -classpath ../deegree2.jar org.deegree.tools.shape.AVL2SLD
```

There is a AVL2SLDtool.sh/.bat in the tools directory which you can adapt to your demands.

4.4.1 Label placement

The deegree WMS supports the automatic and optimized placement of labels for points and polygons. To force an automatic label placement an additional attribute has to be used at the <PointPlacement> element of the <TextSymbolizer>.

```

<sld:UserStyle>
  <sld:Name>default:Springs</sld:Name>
  <sld:Title>default:Springs</sld:Title>
  <sld:IsDefault>1</sld:IsDefault>
  <sld:FeatureTypeStyle>
    <sld:Name>Springs</sld:Name>

```

```

<sld:Rule>
...
</sld:Rule>
<sld:Rule>
  <sld:Name>Springs</sld:Name>
  <sld:MinScaleDenominator>0</sld:MinScaleDenominator>
  <sld:MaxScaleDenominator>500000</sld:MaxScaleDenominator>
  <sld:TextSymbolizer>
    <sld:Geometry>
      <ogc:PropertyName>app:geometry</ogc:PropertyName>
    </sld:Geometry>
    <sld:Label>
      <ogc:PropertyName>app:source</ogc:PropertyName>
    </sld:Label>
    <sld:Font>
      <sld:CssParameter name="font-family">Arial</sld:CssParameter>
      <sld:CssParameter name="font-family">Sans-Serif</sld:CssParameter>
      <sld:CssParameter name="font-style">italic</sld:CssParameter>
      <sld:CssParameter name="font-size">10</sld:CssParameter>
      <sld:CssParameter name="font-color">#222222</sld:CssParameter>
    </sld:Font>
    <sld:LabelPlacement>
      <!-- There are two options to place the text either automatically or
      by setting AnchorPoint/Displacement -->
      <sld:PointPlacement auto="true" />
      <!-- <sld:PointPlacement>
        <sld:Displacement></sld:Displacement>-->
        <!-- The coordinates are given as two floating-point numbers in the
        AnchorPointX and AnchorPointY elements each with values between 0.0
        and 1.0 inclusive. The bounding box of the label to be rendered is
        considered to be in a coordinate space from 0.0 (lower-left corner)
        to 1.0 (upper-right corner), and the anchor position is specified
        as a point in this space. The default point is X=0, Y=0.5, which is
        at the middle height of the left-hand side of the label. -->
        <!-- <sld:AnchorPoint>
          <sld:AnchorPointX>0</sld:AnchorPointX>
          <sld:AnchorPointY>0.5</sld:AnchorPointY>
        </sld:AnchorPoint>-->
        <!-- This will often be used to avoid over-plotting a graphic
        symbol marking a city or some such feature. The displacements are
        in units of pixels above and to the right of the point. A system
        may reflect this displacement about the X and/or Y axes to de-
        conflict labels. The default displacement is X=0, Y=0 -->
        <!--<sld:Displacement>
          <sld:DisplacementX>5</sld:DisplacementX>
          <sld:DisplacementY>-10</sld:DisplacementY>
        </sld:Displacement>
      </sld:PointPlacement>-->
    </sld:LabelPlacement>
    <sld:Halo>
      <sld:Fill>
        <sld:CssParameter name="fill">#0080C0</sld:CssParameter>
        <sld:CssParameter name="fill-opacity">0.6</sld:CssParameter>
      </sld:Fill>
    </sld:Halo>
  </sld:TextSymbolizer>
</sld:Rule>
</sld:FeatureTypeStyle>
</sld:UserStyle>

```

The <UserStyle> given above defines a label for the geometry of a feature named 'app:geometry'. The label will print the content of the features 'app:source' property. The geometry can be a point, a polygon or a multi-polygon. As <LabelPlacement> <PointPlacement> is chosen so the label is related to a point which is the point geometry by itself or the centroid of the visible part of a polygon (<LinePlacement> isn't supported yet). It has an

additional attribute named 'auto' not defined in the SLD specification that forces a label placement optimization if it is set to 'true'. If it is missing or set to 'false' no optimization will be done and the user is free to fix the position of the labels by its own definitions (e.g. AnchorPoint, Displacement, Rotation, ... ; see SLD specifications how to do this).

LabelPlacement for Linestrings is a little bit different, but not much:

```
<sld:Rule>
  <sld:Name>Textsymbolizer</sld:Name>
  <sld:MinScaleDenominator>0</sld:MinScaleDenominator>
  <sld:MaxScaleDenominator>500000</sld:MaxScaleDenominator>
  <sld:TextSymbolizer>
    <sld:Geometry>
      <ogc:PropertyName>app:geometry</ogc:PropertyName>
    </sld:Geometry>
    <sld:Label>
      <ogc:PropertyName>app:code</ogc:PropertyName>
    </sld:Label>
    <sld:Font>
      <CssParameter name="font-family">Serif</CssParameter>
      <CssParameter name="font-style">normal</CssParameter>
      <CssParameter name="font-weight">normal</CssParameter>
      <CssParameter name="font-size">12</CssParameter>
      <CssParameter name="font-color">#000000</CssParameter>
    </sld:Font>
    <sld:LabelPlacement>
      <sld:LinePlacement>
        <sld:PerpendicularOffset>above</sld:PerpendicularOffset>
        <!-- width of the line the label is associated with -->
        <sld:LineWidth>2.5</sld:LineWidth>
        <!-- gap between labels measured in label width -->
        <sld:Gap>5</sld:Gap>
      </sld:LinePlacement>
    </sld:LabelPlacement>
  </sld:TextSymbolizer>
</sld:Rule>
```

Instead of <PointPlacement> the Element <LinePlacement> will be used. At the moment LinePlacement doesn't support automatic label placement optimization but using the three elements contained in the <LinePlacement> element you can already realize good results. The <PerpendicularOffset> defines the relative position of the label according to the line. You can use a number as defined in the SLD specifications or set a predefined value (above, below, center). If you use one of predefined values it is useful to set a value for <LineWidth> to ensure that if you choose 'above' the label will be really above the line and not partially within it.

Using the <Gap> element you can determine the gap between the labels of one Linestring. The value of <Gap> is measured in the length of the label string. This means if a river shall be labeled with 'Rhine' one gap will have the length of the string 'Rhine'. The labeling will look like this Rhine Rhine Rhine. If you choose <Gap>0</Gap> you will get a labeling like this Rhine Rhine Rhine Rhine Rhine.

5 Advanced configuration

5.1 Manual Tomcat integration

The location of deegree's libraries and the central deegree wms configuration file `wms_configuration.xml` should be registered with the Servlet Engine (in this case Apache Tomcat 5.5). Tomcat offers several possibilities to register and configure web contexts.

The easiest way to register deegree web services with Tomcat is to copy the `deegree-wms.war` file to the `$TOMCAT_HOME$/webapps` directory. You can do this either with running or stopped tomcat. If the tomcat is started afterwards, the application should be automatically deployed. Tomcat will unpack the `deegree-wms.war` file (which is nothing more than a .zip file) to the webapps directory. The name of the .war sets the name of the service address:

`http://localhost:8080/deegree-wms`

If you want to do the Tomcat installation process manually use the steps described in the following.

Unpack the `deegree-wms.war` to a directory of your choice (e.g. `c:/deegree/webapps/deegree-wms`).

Afterwards Tomcat needs information about the root directory of the WMS. The easiest way is to create an XML-file in the directory `$TOMCAT_HOME$/conf/Catalina/localhost`, named exactly as the service e.g. `deegree-wms.xml`, and fill it with the following information

```
<Context docBase="c:/deegree/webapps/deegree-wms" path="/deegree-wms">
</Context>
```

where the `docBase` attribute reflects the physical location of the deegree service in the file system and the `path` attribute describes the virtual location of the main directory of the deegree web service. In the example, the root directory of the service is accessible at `http://my.server.domain/deegree-wms/`. For further information have a look at the Tomcat documentation included in the installation.

The name of the deegree-service directory is arbitrary whereas Tomcat definitely looks for a subdirectory `WEB-INF` (in capital letters – even on a Windows system) in the root directory. You will find this directory after tomcat automatically unpacked the war archive. Here the `Deployment-Descriptor` (`web.xml`) is located, which is analysed by Tomcat to identify the servlet(s) belonging to the application, their names, the parameters that are delivered to the servlet(s) and information about the existing access restrictions.

Before starting deegree WMS the following dataset entry in `web.xml` is essential:

```
<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
```

```
<web-app>
  <display-name>deegree 2.2</display-name>
  <description>deegree 2.2 OWS</description>
  <servlet>
    <servlet-name>owservice</servlet-name>
    <servlet-class>org.deegree.enterprise.servlet.OGCServletController</servlet-
    class>

    <init-param>
      <param-name>services</param-name>
      <param-value>wms</param-value>
      <description>
        list of supported services, e.g.: wfs,wms,wcs (comma separated) always use
        lowercase
      </description>
    </init-param>

    <!-- WMS INITIALIZING PARAMETERS -->
    <init-param>
      <param-name>wms.handler</param-name>
      <param-value>org.deegree.enterprise.servlet.WMSHandler</param-value>
    </init-param>
    <init-param>
      <param-name>wms.config</param-name>
      <param-value>WEB-INF/conf/wms/wms_configuration.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>owservice</servlet-name>
    <url-pattern>/services</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>/index.jsp</welcome-file>
  </welcome-file-list>
  <error-page>
    <error-code>500</error-code>
    <location>/error.jsp</location>
  </error-page>
  <error-page>
    <exception-type>org.deegree.ogcwebservices.OGCWebServiceException</exception-
    type>
    <location>/error.jsp</location>
  </error-page>
</web-app>
```

The name of the servlet and of the java-class representing the servlet should be indicated in the <servlet> tags. The servlet-name can be user defined, but care should be taken that the same name that is defined here is also used in the servlet-mapping. The servlet is located in the deegree2.jar library.

The tag <init-param> defines parameters that are analyzed by the servlet, during initialization. The transferred parameters are

- 'services': The value of this parameter contains a comma separated list of OWS that will be made available through the context. In the example only a 'wms' is defined to be available (other possible values at the moment are: wfs, wcs, sos, wpvs and csw).
- For each service listed in the 'service' init-param a handler class and a configuration file must be referenced.

- The name of the init-param for defining the handler starts with the service name ('wms' in the example) followed by '.handler'. The value of this parameter is the name of the handler class to be used. It is possible to write a different class for this and reference it accordingly. As default 'org.deegree.enterprise.servlet.WMSHandler' should be used.
- The name of the init-param for defining the main configuration file of a service also starts with the service name followed by '.config'. Note that you can use a relative path to the configuration file starting at the WEB-INF directory of the context.

If you want to make more than one service available through a servlet context, web.xml looks like this (the example defines a 'wms' as well as a 'wfs' and if you uncomment the WCS section even this one is accessible):

```
<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <display-name>deegree 2.2</display-name>
  <description>deegree 2.2 OWS</description>
  <servlet>
    <servlet-name>owservice</servlet-name>
    <servlet-class>org.deegree.enterprise.servlet.OGCServletController</servlet-
class>

    <init-param>
      <param-name>services</param-name>
      <param-value>wms,wfs</param-value>
      <description>
        list of supported services, e.g.: wfs,wms,wcs (comma separated) always use
        lowercase
      </description>
    </init-param>

    <!-- WMS INITIALIZING PARAMETERS -->
    <init-param>
      <param-name>wms.handler</param-name>
      <param-value>org.deegree.enterprise.servlet.WMSHandler</param-value>
    </init-param>
    <init-param>
      <param-name>wms.config</param-name>
      <param-value>WEB-INF/conf/wms/wms_configuration.xml</param-value>
    </init-param>

    <!-- WFS INITIALIZING PARAMETERS -->
    <init-param>
      <param-name>wfs.handler</param-name>
      <param-value>org.deegree.enterprise.servlet.WFSHandler</param-value>
    </init-param>
    <init-param>
      <param-name>wfs.config</param-name>
      <param-value>WEB-INF/conf/wms/LOCALWFS_configuration.xml</param-value>
    </init-param>

    <!-- WCS INITIALIZING PARAMETERS -->
    <!--
    <init-param>
      <param-name>wcs.handler</param-name>
      <param-value>org.deegree.enterprise.servlet.WCSHandler</param-value>
    </init-param>
    <init-param>
      <param-name>wcs.config</param-name>
```



```

    <param-value>WEB-INF/conf/wcs/LOCALWCS_configuration.xml</param-value>
  </init-param>
-->
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>owservice</servlet-name>
  <url-pattern>/services</url-pattern>
</servlet-mapping>
<welcome-file-list>
  <welcome-file>/index.jsp</welcome-file>
</welcome-file-list>
<error-page>
  <error-code>500</error-code>
  <location>/error.jsp</location>
</error-page>
<error-page>
  <exception-type>org.deegree.ogcwebservices.OGCWebServiceException</exception-
  type>
  <location>/error.jsp</location>
</error-page>
</web-app>

```

The tag `<servlet-mapping>` defines the alias name for the servlet. It is not necessary that the `<servlet-name>` and `<url-pattern>` are identical. `<url-pattern>` is the name for the parameter the servlet will be called through (part of the base-URL of the service that all requests have to use). Combined with the path settings in `$TOMCAT_HOME$/conf/Catalina/localhost/deegree-wms.xml` and respectively the name of the .war which you deployed in the `$TOMCAT_HOME$/webapps` (in our example `deegree-wms`) you should be able to point to your WMS (OWS) via the following URL: `http://my.server.domain/deegree-wms/services?`

5.2 Offer own vector data (short description)

This deegree 2.2 WMS release comes bundled with a configuration of WFS and WCS as these are the data sources of the WMS. It is not possible for the WMS to connect to databases or files directly. You will find detailed information on how to configure the deegree 2.2 WFS and 2.2 WCS in separate download packages/documentation. But still you find some scripts under `wms_home/WEB-INF/conf/tools` with some short hints to assist you to add own data. The three steps you have to take are

1. create a `featuretypedefinition.xsd` for the LOCALWFS and place it in `wms_home/WEB-INF/conf/wms/featuretypes`
2. Adapt the `wms_home/WEB-INF/conf/wms/wms_configuration.xml` to your demands (add needed SRS support, bounding boxes, etc) and add a layer definition
3. create a `style.xml` or use an existing file. Create a `<UserStyle>` and reference it in the `wms_configuration.xml` (refer Chapter 4.2 et seqq)
4. restart tomcat and you are ready!

Featuretype definitions can be created with support by tools. Tools are supplied for Windows (xxx.bat) and Linux (xxx.sh) operating systems. Before you start one of the tools you have to adapt them to your demands. You find preconfigured commands for the demo datasets which you can uncomment and adapt. You can edit these scripts with any text editor. Afterwards just go with your shell/console to the mentioned tools directory and run the desired script. To create a featuretype definition on shape files use the script 'ShapetoFeatureTypeDef', to create Postgis database access use the script DBtoFeatureTypeDef.

Open the created featuretypedefinition.xsd and do further manual configuration:

- set the CRS to EPSG:yourCode in the tag `<deegree:DefaultSRS> EPSG:26912</deegree:DefaultSRS>`
- check if the path to the shapefile is correct. Relative paths must start at the featuretypedefinition.xsd!
- if a database is used which stores the used CRS, then specify the table `srs=' '` (without 'EPSG:' !). If no CRS is specified in DB or if used for shapefile access then leave `srs='-1'`
- ```
<xsd:element name='geometry' type='gml:GeometryPropertyType'>
 <xsd:annotation>
 <xsd:appinfo>
 <deegree:Content>
 <deegree:MappingField field='the_geom' type='GEOMETRY'
 srs='26912' />
 </deegree:Content>
 </xsd:appinfo>
 </xsd:annotation>
</xsd:element>
```
- see `Commented_FeaturetypeDefiniton.xsd.txt` under `$wms_home$/WEB-INF/conf/wms/featuretypes` for further details
- restart tomcat and the WFS should deliver this feature type.

Follow with the steps 2 & 3 and you are ready.

## Appendix A Example wms\_configuration.xml

Path of file: \$wms\_home\$/WEB-INF/conf/wms/

```
<?xml version="1.0" encoding="UTF-8"?>
<WMT_MS_Capabilities xmlns:deegree="http://www.deegree.org/wms"
xmlns:sld="http://www.opengis.net/sld" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1.1" updateSequence="1.1.0">
 <deegree:DeegreeParam>
 <!--
 The online resource parameter must be set here: The 'deegree-wms'
 originates form the context file in apache under
 $Tomcat_home$/conf/Catalina/localhost/deegree-wms.xml or the name of
 the .war file under $Tomcat_home$/webapps/ depending on where the root
 directory of the service is located; the services url-pattern originates
 from the deployment descriptor under wms_home/WEB-INF/web.xml
 <url-pattern>/services</url-pattern>
 -->
 <deegree:DefaultOnlineResource xlink:type="simple"
 xlink:href="http://localhost:8080/deegree-wms/services" />
 <!-- default = 100 (MB) -->
 <deegree:CacheSize>100</deegree:CacheSize>
 <!-- maximum lifetime of the processes in the WMS; default = 3600 (sec)-->
 <deegree:MaxLifeTime>3600</deegree:MaxLifeTime>
 <!-- maximum time for the execution of a request until an exception of
 time-exceed is thrown. default is 15 seconds -->
 <deegree:RequestTimeLimit>45</deegree:RequestTimeLimit>
 <!-- determines the quality of the map/image generated from a GetMap
 request. the parameter only will be evaluated if the desired format is
 able to handle it. the range of values is 0 ... 1, where 1 is best and
 0 is worst. default is 0.95 -->
 <deegree:MapQuality>0.95</deegree:MapQuality>
 <!-- maximum map width that can be requested. default = 1000 -->
 <deegree:MaxMapWidth>1500</deegree:MaxMapWidth>
 <!-- maximum map height that can be requested. default = 1000 -->
 <deegree:MaxMapHeight>1500</deegree:MaxMapHeight>
 <deegree:AntiAliased>true</deegree:AntiAliased>
 <!--
 copyright note that will be drawn to the left bottom side of the maps;
 you can also reference a graphic file using absolute path to file e.g.
 c:/images/mylogo.jpg
 -->
 <deegree:Copyright>deegree-WMS v2.3 2010</deegree:Copyright>
 <!--
 radius of the circle around the point a user has clicked to that will be
 considered for creating a search area for a GetFeatureInfo request;
 default = 5
 -->
 <deegree:FeatureInfoRadius>10</deegree:FeatureInfoRadius>
 <!--
 returns the URL where the DTD for OGC WMS capabilities DOCTYPE definiton
 is located. default =
 http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd
 -->
 <!--
 <deegree:DTDLocation>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd"/>
 </deegree:DTDLocation>
 -->
 <!--
 define a proxy server if the WMS is seperated by a firewall from the DTD-
 location or the locations of connected REMOTEWFS, REMOTEWCS and
 REMOTEWMS as an alternative you can set commandline parameters at the
 call of the java interpreter like this:
 java -DproxyHost=131.220.106.104 -DproxyPort=1234 -classpath ...
 -->
 <!--
```

```

<Proxy proxyHost="131.220.106.104" proxyPort="1234"/>
-->
<!-- default color depth for GetMap which will be used for 'image/png';
default is image/png; mode=24bit -->
<degree:DefaultPNGFormat>image/png; mode=24bit</degree:DefaultPNGFormat>
</degree:DegreeParam>
<Service>
 <Name>OGC:WMS</Name>
 <Title>degree wms</Title>
 <!-- abstract and keywords are optional -->
 <Abstract>wms reference implementation</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>wms</Keyword>
 </KeywordList>
 <!-- The online resource must be specified here -->
 <OnlineResource xlink:type="simple"
xlink:href="http://localhost:8080/degree-wms/services" />
 <!-- the following service information is optional -->
 <ContactInformation>
 <ContactPersonPrimary>
 <ContactPerson>Andreas Poth</ContactPerson>
 <ContactOrganization>lat/lon</ContactOrganization>
 </ContactPersonPrimary>
 <ContactPosition>Technical Director</ContactPosition>
 <ContactAddress>
 <AddressType>XXXX</AddressType>
 <Address>Aennchenstr. 19</Address>
 <City>Bonn</City>
 <StateOrProvince>NRW</StateOrProvince>
 <PostCode>53177</PostCode>
 <Country>Germany</Country>
 </ContactAddress>
 <ContactVoiceTelephone>0049228184960</ContactVoiceTelephone>
 <ContactFacsimileTelephone>00492281849629</ContactFacsimileTelephone>
 <ContactElectronicMailAddress>info@latlon.de
 </ContactElectronicMailAddress>
 </ContactInformation>
 <Fees>none</Fees>
 <AccessConstraints>none</AccessConstraints>
</Service>
<Capability>
 <Request>
 <!-- by default available -->
 <GetCapabilities>
 <!-- default and mandatory = application/vnd.ogc.wms_xml -->
 <Format>application/vnd.ogc.wms_xml</Format>
 <DCPType>
 <HTTP>
 <Get>
 <!-- If left empty, it will be automatically filled with the
parameters of <degree:DefaultOnlineResource> -->
 <!--
 <OnlineResource xlink:type="simple"
xlink:href="http://localhost:8080/degree-wms/services?" />
 -->
 </Get>
 <!-- POST isn't supported -->
 </HTTP>
 </DCPType>
 </GetCapabilities>
 <!-- by default available -->
 <GetMap>
 <!-- default = image/gif; image/png; image/jpeg -->
 <!-- mandatory = image/png -->
 <Format>image/gif</Format>
 <Format>image/png</Format>
 <Format>image/png; mode=8bit</Format>
 <Format>image/png; mode=24bit</Format>

```

```

<Format>image/jpg</Format>
<Format>image/jpeg</Format>
<Format>image/tif</Format>
<Format>image/bmp</Format>
<DCPType>
 <HTTP>
 <Get>
 <!-- If left empty, it will be automatically filled with the
 parameters of <degree:DefaultOnlineResource> -->
 <!--
 <OnlineResource xlink:type="simple"
 xlink:href="http://localhost:8080/degree-wms/services?" />
 -->
 </Get>
 <Post>
 <!-- If left empty, it will be automatically filled with the
 parameters of <degree:DefaultOnlineResource> -->
 <!--
 <OnlineResource xlink:type="simple"
 xlink:href="http://localhost:8080/degree-wms/services" />
 -->
 </Post>
 </HTTP>
</DCPType>
</GetMap>
<!-- by default available -->
<GetFeatureInfo>
 <!-- default & mandatory = application/vnd.ogc.gml -->
 <Format>application/vnd.ogc.gml</Format>
 <Format>text/plain</Format>
 <Format>text/html</Format>
 <DCPType>
 <HTTP>
 <Get>
 <!-- If left empty, it will be automatically filled with the
 parameters of <degree:DefaultOnlineResource> -->
 <!--
 <OnlineResource xlink:type="simple"
 xlink:href="http://localhost:8080/degree-wms/services?" />
 -->
 </Get>
 <!-- POST isn't supported -->
 </HTTP>
 </DCPType>
</GetFeatureInfo>
<GetLegendGraphic>
 <!-- default = image/gif; image/png; image/jpg -->
 <!-- mandatory = image/png -->
 <Format>image/gif</Format>
 <Format>image/png</Format>
 <Format>image/jpeg</Format>
 <Format>image/jpg</Format>
 <Format>image/tif</Format>
 <Format>image/bmp</Format>
 <DCPType>
 <HTTP>
 <Get>
 <!-- If left empty, it will be automatically filled with the
 parameters of <degree:DefaultOnlineResource> -->
 <!--
 <OnlineResource xlink:type="simple"
 xlink:href="http://localhost:8080/degree-wms/services?" />
 -->
 </Get>
 <!-- POST isn't supported -->
 </HTTP>
 </DCPType>
</GetLegendGraphic>
</Request>

```

```

<Exception>
 <!-- default & mandatory= application/vnd.ogc.se_xml -->
 <Format>application/vnd.ogc.se_xml</Format>
 <Format>application/vnd.ogc.se_inimage</Format>
 <Format>application/vnd.ogc.se_blank</Format>
</Exception>
<!-- optional; default = all false -->
<UserDefinedSymbolization SupportSLD="1" UserLayer="1" UserStyle="1"
RemoteWFS="1" />
<!-- cascaded is optional and default=false -->
<!-- TOP LAYER
As you are able to nest Layers as in this example please always make sure
that you have the appropriate number
of opening layer tags (<Layer>) and closing layer tags (</Layer>)
-->
<Layer queryable="0" cascaded="0" noSubsets="0"
 xmlns:app="http://www.deegree.org/app">
 <Title>deegree 2.3 Demo WMS</Title>
 <!-- abstract and keyword are optional -->
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!-- Just add the <SRS> you wish to support with your WMS -->
 <!-- default = EPSG:4326 -->
 <SRS>EPSG:4326</SRS>
 <SRS>EPSG:26912</SRS>
 <SRS>EPSG:25832</SRS>
 <SRS>EPSG:31467</SRS>
 <!--
 It is strongly recommended that you set a LatLonBoundingBox; further
 <BoundingBox>es are optional.
 Each layer inherits the BBoxes of the parent layer as long as no
 explicit on is set.
 -->
 <LatLonBoundingBox miny="-90" maxy="90" minx="-180" maxx="180" />
 <!-- Setting <BoundingBox>es ist optional but if set be sure to set the
 correct SRS="EPSG:" -->
 <BoundingBox SRS="EPSG:26912" miny="3581352" maxy="5432672" minx="0"
 maxx="1504379" />
 <BoundingBox SRS="EPSG:4326" miny="-90" maxy="90" minx="-180"
 maxx="180" />
 <!-- default 0 ; 9E99 -->
 <ScaleHint min="0" max="10000000" />
 <!--
 This nested layer has a <Name>, but no datasource. You can request this
 layer anyway as you get all the sub-layers. You even get the
 GetFeatureInfo but your client must be able to handle the result as
 each sublayer has it's own properties (attributes). If you don't want
 this layer to be accessible remove the name tag
 (<Name>citelayers</Name>).
 -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>citelayers</Name>
 <Title>Cite Layers</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <LatLonBoundingBox miny="-1" maxy="1" minx="-1" maxx="1" />

```

```

<BoundingBox SRS="EPSG:4326" miny="-1" maxy="1" minx="-1" maxx="1"
/>
<!--
 queryable="1" sets whether GetFeatureinfo is enabled or not. for
 this layer if set to "1" at least one datasource must be set to
 "1" too
-->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
<!-- please choose a unique <name> without spaces and a title which
can contain more details -->
 <Name>cite:BasicPolygons</Name>
 <Title>cite:BasicPolygons</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <!-- sets the range of WMS scale, where the WMS will serve data
 for this layer -->
 <MinScaleDenominator>0</MinScaleDenominator>
 <MaxScaleDenominator>5000</MaxScaleDenominator>
 <!-- queryable="1" sets whether GetFeatureinfo is enabled or
 not for this datasource -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:BasicPolygons</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <!--
 name of the property that contains the geometries for this
 layer
 this element will only be solved if the data source is a
 LOCALWFS or a REMOTEWFS
 default = 'app:GEOM'
 -->
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <!--
 set the name of the style and the location
 <deegree:StyleResource> where the WMS will find a
 style definition with the same <UserStyle> name
 -->
 <Name>default:cite:BasicPolygons</Name>
 <Title>default:cite:BasicPolygons</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 <!-- If LegendURL is set for layers the default automated
 legend graphic will be overwritten -->
 <!--
 <LegendURL width="50" height="50">
 <Format>image/jpg</Format>
 <OnlineResource xlink="http://www.w3.org/1999/xlink"
type="simple" xlink:href="http://www.yourdomain.com/legend/teststyle.jpg"/>
 </LegendURL>
 -->
 </Style>
</Layer>
<!-- The following Layer definition could be used as default for
your own layers -->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">

```

```

<Name>cite:Bridges</Name>
<Title>cite:Bridges</Title>
<Abstract>deegree demo WMS</Abstract>
<KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
</KeywordList>
<!--
<MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
</MetadataURL>
-->
<deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Bridges</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
</deegree:DataSource>
<Style>
 <Name>default:cite:Bridges</Name>
 <Title>default:cite:Bridges</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:Buildings</Name>
 <Title>cite:Buildings</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Buildings</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:Buildings</Name>
 <Title>default:cite:Buildings</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:BuildingCenters</Name>
 <Title>cite:BuildingCenters</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">

```



```

 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Buildings</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:BuildingCenters</Name>
 <Title>default:cite:BuildingCenters</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:DividedRoutes</Name>
 <Title>cite:DividedRoutes</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:DividedRoutes</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:DividedRoutes</Name>
 <Title>default:cite:DividedRoutes</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:Forests</Name>
 <Title>cite:Forests</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Forests</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>

```

```

 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geom</degree:GeometryProperty>
</degree:DataSource>
<Style>
 <Name>default:cite:Forests</Name>
 <Title>default:cite:Forests</Title>
 <degree:StyleResource>styles/cite.xml</degree:StyleResource>
</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:Lakes</Name>
 <Title>cite:Lakes</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Lakes</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geom</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:cite:Lakes</Name>
 <Title>default:cite:Lakes</Title>
 <degree:StyleResource>styles/cite.xml</degree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:MapNeatline</Name>
 <Title>cite:MapNeatline</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:MapNeatline</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geom</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:cite:MapNeatline</Name>
 <Title>default:cite:MapNeatline</Title>
 <degree:StyleResource>styles/cite.xml</degree:StyleResource>
 </Style>
</Layer>

```

```

</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:NamedPlaces</Name>
 <Title>cite:NamedPlaces</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:NamedPlaces</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:NamedPlaces</Name>
 <Title>default:cite:NamedPlaces</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:Ponds</Name>
 <Title>cite:Ponds</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Ponds</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:Ponds</Name>
 <Title>default:cite:Ponds</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:RoadSegments</Name>
 <Title>cite:RoadSegments</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>

```

```

</KeywordList>
<!--
<MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
</MetadataURL>
-->
<deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:RoadSegments</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
</deegree:DataSource>
<Style>
 <Name>default:cite:RoadSegments</Name>
 <Title>default:cite:RoadSegments</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>cite:Streams</Name>
 <Title>cite:Streams</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Streams</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geom</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:cite:Streams</Name>
 <Title>default:cite:Streams</Title>
 <deegree:StyleResource>styles/cite.xml</deegree:StyleResource>
 </Style>
</Layer>
</Layer>

<!-- ++++++ -->
<!-- ++++++ Utah ++++++ -->
<!-- ++++++ -->

<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Utah Demo Data</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>

```

```

 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <!-- Example for a LOCALWCS -->
 <Layer queryable="0" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>orthoimage</Name>
 <Title>Ortho Imagery of Salt Lake City</Title>
 <SRS>EPSG:26912</SRS>
 <SRS>EPSG:4326</SRS>
 <Layer queryable="0" noSubsets="0" fixedWidth="0"
 fixedHeight="0">
 <Name>orthoimage1</Name>
 <Title>Ortho Imagery of Salt Lake City no Bbox</Title>
 <SRS>EPSG:26912</SRS>
 <SRS>EPSG:4326</SRS>
 <LatLonBoundingBox miny="40.719021928792515"
 maxy="40.82820182642026" minx="-111.94721360400128"
 maxx="-111.80644038553243" />
 <BoundingBox SRS="EPSG:26912" miny="4507998.0"
 maxy="4519999.0" minx="420000.0" maxx="432001.0" />
 <BoundingBox SRS="EPSG:4326" miny="40.719021928792515"
 maxy="40.82820182642026" minx="-111.94721360400128"
 maxx="-111.80644038553243" />
 <deegree:DataSource failOnException="1" queryable="0">
 <deegree:Name>saltlakesatellite</deegree:Name>
 <deegree:Type>LOCALWCS</deegree:Type>
 <deegree:ScaleHint min="0.0" max="100" />
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWCS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:FilterCondition>
 <deegree:WCSRequest>
 <!--
CDATA[VERSION=1.0.0&coverage=saltlakesatellite&TRANSPARENT=TRUE&FORMAT=jpg&EXCEPTIO
NS=application/vnd.ogc.se_xml]]></deegree:WCSRequest>
 </deegree:FilterCondition>
 </deegree:DataSource>
 </deegree:DataSource>
 </deegree:DataSource>
 <Layer queryable="0" noSubsets="0" fixedWidth="0"
 fixedHeight="0">
 <Name>bbox</Name>
 <Title>BBox of Ortho Imagery of Salt Lake City</Title>
 <SRS>EPSG:26912</SRS>
 <SRS>EPSG:4326</SRS>
 <deegree:DataSource failOnException="1" queryable="0">
 <deegree:Name>app:Bbox_ortho_image</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 <deegree:ScaleHint min="100" max="50000000" />
 </deegree:DataSource>
 <Style>
 <Name>default:bbox</Name>
 <Title>default:bbox</Title>
 </deegree:StyleResource>styles/utah_bbox.xml</deegree:StyleResource>
 </Style>
 </Layer>
 </Layer>
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>StateBoundary</Name>
 <Title>StateBoundary</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>

```

```

 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <LatLonBoundingBox miny="36" maxy="43" minx="-115" maxx="-108" />
 <ScaleHint min="0" max="2800" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:StateBoundary</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:StateBoundary</Name>
 <Title>default:StateBoundary</Title>
 </Style>
<degree:StyleResource>styles/utah_state.xml</degree:StyleResource>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>StateOverview</Name>
 <Title>StateOverview</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:StateBoundary</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
 </degree:DataSource>
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:CountyBoundaries_edited</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:StateOverview</Name>
 <Title>default:StateOverview</Title>
 </Style>
<degree:StyleResource>styles/utah_state.xml</degree:StyleResource>
</Layer>

```

```

<!-- Example for a REMOTEWMS; cascaded="1" indicates, that this
 layer originates from a remote WMS -->
<!--
<Layer queryable="1" cascaded="1" noSubsets="0" fixedWidth="0"
fixedHeight="0">
 <Name>landcover</Name>
 <Title>National Land Cover Dataset (JUST EPSG:4326 => WGS
 84)</Title>
 <SRS>EPSG:4326</SRS>
 <LatLonBoundingBox miny="37.88970184326172"
 maxy="42.91960144042969" minx="-115.55139923095703"
 maxx="-110.41089630126953" />
 <degree:DataSource failOnException="1" queryable="1">
 <degree:Name>3:5</degree:Name>
 <degree:Type>REMOTEWMS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource
xlink:type="simple"link:href="http://columbo.nrlssc.navy.mil/ogcwms/servlet/WMSServ
let/Utah_State_University_Water_Initiative.wms?
SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities" />
 </degree:OWSCapabilities>
 <degree:FilterCondition>
 <degree:WMSRequest><![
CDATA[Version=1.1.1&FORMAT=image/png&TRANSPARENT=true&EXCEPTIONS=application/vnd.o
gc.se_inimage&BGCOLOR=0xffffffff&LAYERS=3:5&STYLES=]]></degree:WMSRequest>
 </degree:FilterCondition>
 <degree:ScaleHint min="0.0" max="9999999999" />
 <degree:TransparentColor>
 <degree:Color>#000000</degree:Color>
 <degree:Color>#FF0000</degree:Color>
 </degree:TransparentColor>
 </degree:DataSource>
</Layer>
-->
<!-- This layer contains several sublayers and is able load these
 as one as it has a <name> -->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Administration</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 <!-- This layer contains 3 style defintions -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0"
fixedHeight="0">
 <Name>Counties</Name>
 <Title>County Boundaries</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:CountyBoundaries_edited</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>

```

```

 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
</degree:DataSource>
<Style>
 <Name>default:Counties</Name>
 <Title>default:Counties</Title>
<degree:StyleResource>styles/utah_county.xml</degree:StyleResource>
</Style>
<Style>
 <Name>GreyCounties</Name>
 <Title>GreyCounties</Title>
<degree:StyleResource>styles/utah_county.xml</degree:StyleResource>
</Style>
<Style>
 <Name>ColourfulCounties</Name>
 <Title>ColourfulCounties</Title>
<degree:StyleResource>styles/utah_county.xml</degree:StyleResource>
</Style>
</Layer>
<!-- This Layer contains 2 style definitions -->
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>ZipCodes</Name>
 <Title>Zip Codes</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:ZipCodes</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:ZipCodes</Name>
 <Title>default:ZipCodes</Title>
 <degree:StyleResource>styles/utah_state.xml</degree:StyleResource>
 </Style>
 <Style>
 <Name>ZipCodesPop</Name>
 <Title>ZipCodes Population</Title>
 <degree:StyleResource>styles/utah_state.xml</degree:StyleResource>
 </Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Municipalities</Name>
 <Title>Municipalities</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">

```



```

 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Municipalities_edited</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 </deegree:DataSource>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 </deegree:DataStore>
 <Style>
 <Name>default:Municipalities</Name>
 <Title>default:Municipalities</Title>
 </deegree:StyleResource>styles/utah_state.xml</deegree:StyleResource>
 </Style>
</Layer>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Geology</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <!--
 This layer contains OGC filter encoding. The WMS collects just
 defined WFS properties (attributes)
 as well as selected content (COAL*,PC,GF)
 -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>EnergyResources</Name>
 <Title>Energy Resources</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:EnergyResources</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 </deegree:DataSource>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 <deegree:FilterCondition>
 <wfs:Query typeName="app:EnergyResources"
 xmlns:gml="http://www.opengis.net/gml"

```

```

 xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:wfs="http://www.opengis.net/wfs"
 <wfs:PropertyName>app:code</wfs:PropertyName>
 <!--
 <wfs:PropertyName>app:shape_area</wfs:PropertyName> -->
 <wfs:PropertyName>app:shape_len</wfs:PropertyName>
 <ogc:Filter>
 <ogc:Or>
 <ogc:Or>
 <ogc:PropertyIsEqualTo>
 <ogc:PropertyName>app:code</ogc:PropertyName>
 <ogc:Literal>PC</ogc:Literal>
 </ogc:PropertyIsEqualTo>
 <ogc:PropertyIsEqualTo>
 <ogc:PropertyName>app:code</ogc:PropertyName>
 <ogc:Literal>GF</ogc:Literal>
 </ogc:PropertyIsEqualTo>
 </ogc:Or>
 </ogc:Or>
 </ogc:Filter>
 <ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\ ">
 <ogc:PropertyName>app:code</ogc:PropertyName>
 <ogc:Literal>COAL*</ogc:Literal>
 </ogc:PropertyIsLike>
 </ogc:Or>
 </ogc:Filter>
</wfs:Query>
</degree:FilterCondition>
</degree:DataSource>
<Style>
 <Name>default:EnergyResources</Name>
 <Title>default:EnergyResources</Title>
<degree:StyleResource>styles/utah_geology.xml</degree:StyleResource>
</Style>
</Layer>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Hydrology</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <!-- Layer containing points -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Springs</Name>
 <Title>Springs of Utah</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="560" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Springs</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>

```

```

 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
<degree:GeometryProperty>app:geometry</degree:GeometryProperty>
</degree:DataSource>
<Style>
 <Name>default:Springs</Name>
 <Title>default:Springs</Title>
 degree:StyleResource>styles/utah_hydrology.xml</degree:StyleResource>
</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Lake</Name>
 <Title>Lakes of Utah</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2100" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Lake</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 <degree:GeometryProperty>app:geometry</degree:GeometryProperty>
 </degree:DataSource>
 <Style>
 <Name>default:Lake</Name>
 <Title>default:Lake</Title>
 <degree:StyleResource>styles/utah_hydrology.xml</degree:StyleResource>
 </Style>
 </Layer>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Land Use</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Vegetation</Name>
 <Title>Dominant Vegetation</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>

```

```

 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Vegetation</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 </deegree:DataSource>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 </deegree:DataResource>
 <Style>
 <Name>default:Vegetation</Name>
 <Title>default:Vegetation</Title>
 </deegree:StyleResource>
 </Style>
</Layer>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Infrastructure</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Airports</Name>
 <Title>Airports of Utah</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Airports</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 </deegree:DataSource>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 </deegree:DataResource>
 <Style>
 <Name>default:Airports</Name>
 <Title>default:Airports</Title>
 </deegree:StyleResource>
 </Style>
 </Layer>
 <!-- This layer has 2 datasources used for different scales -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Roads</Name>

```

```

<Title>Roads of Utah</Title>
<Abstract>deegree demo WMS</Abstract>
<KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
</KeywordList>
<!--
<MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
</MetadataURL>
-->
<MinScaleDenominator>0</MinScaleDenominator>
<MaxScaleDenominator>10000000</MaxScaleDenominator>
<deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Roads</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
<deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 <deegree:ScaleHint min="0.0" max="50" />
</deegree:DataSource>
<deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Roads500</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
<deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 <deegree:ScaleHint min="50.0" max="2000000000" />
</deegree:DataSource>
<Style>
 <Name>default:Roads</Name>
 <Title>default:Roads</Title>
<deegree:StyleResource>styles/utah_roads.xml</deegree:StyleResource>
</Style>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>Railroads</Name>
 <Title>Railroads of Utah</Title>
 <Abstract>deegree demo WMS</Abstract>
 <KeywordList>
 <Keyword>deegree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.deegree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <deegree:DataSource failOnException="0" queryable="1">
 <deegree:Name>app:Railroads</deegree:Name>
 <deegree:Type>LOCALWFS</deegree:Type>
 <deegree:OWSCapabilities>
 <deegree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </deegree:OWSCapabilities>
 <deegree:GeometryProperty>app:geometry</deegree:GeometryProperty>
 </deegree:DataSource>
 <Style>
 <Name>default:Railroads</Name>
 <Title>default:Railroads</Title>

```

```

<degree:StyleResource>styles/utah_infra.xml</degree:StyleResource>
</Style>
</Layer>
</Layer>
<Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Title>Elevation</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <!--
 This layer has 3 datasources used for different scale ranges;
 additionally the geometry property is
 renamed in the featuretype definition file to contourLine which
 also has to be set in the featuretypedefinition
 -->
 <Layer queryable="1" noSubsets="0" fixedWidth="0" fixedHeight="0">
 <Name>ElevationContours</Name>
 <Title>Elevation Contours</Title>
 <Abstract>degree demo WMS</Abstract>
 <KeywordList>
 <Keyword>degree</Keyword>
 <Keyword>layer</Keyword>
 </KeywordList>
 <!--
 <MetadataURL type="ISO19115:2003">
 <Format>text/html</Format>
 <OnlineResource xlink:type="simple"
 xlink:href="http://www.degree.org" />
 </MetadataURL>
 -->
 <ScaleHint min="0" max="2800" />
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Contours500Ft</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:contourLine</degree:GeometryProperty>
 <degree:ScaleHint min="0.0" max="150" />
 </degree:DataSource>
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Contours1000Ft</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:contourLine</degree:GeometryProperty>
 <degree:ScaleHint min="150" max="500" />
 </degree:DataSource>
 <degree:DataSource failOnException="0" queryable="1">
 <degree:Name>app:Contours2500Ft</degree:Name>
 <degree:Type>LOCALWFS</degree:Type>
 <degree:OWSCapabilities>
 <degree:OnlineResource xlink:type="simple"
 xlink:href="LOCALWFS_configuration.xml" />
 </degree:OWSCapabilities>
 </degree:DataSource>
 <degree:GeometryProperty>app:contourLine</degree:GeometryProperty>

```

```

 <degree:ScaleHint min="500" max="200000000" />
 </degree:DataSource>
 <Style>
 <Name>default:ElevationContours</Name>
 <Title>default:ElevationContours</Title>
 <degree:StyleResource>styles/utah_general.xml</degree:StyleResource>
 </Style>
</Layer>
</Layer>
<!-- ++++++----->
<!-- ++++++ Please Note: ++++++----->
<!-- ++++++ The Demo wms_configuration.xml ++++++----->
<!-- ++++++ contains more than these example layers.+ -->
<!-- ++++++ The example just ends here, ++++++----->
<!-- ++++++ to make documentation not too big ++++++----->
<!-- ++++++----->
 </Layer>
</Capability>
</WMT_MS_Capabilities>

```

## Appendix B Supported SRS

Change to `$wms_home$/WEB-INF/scripts` and execute:

Linux:

```
java -cp ../../lib/deegree2.jar:../../lib/jai/jai_core.jar
org.deegree.tools.srs.SRSInfo
```

Windows

```
java -cp ../../lib\deegree2.jar;../../lib\jai\jai_core.jar
org.deegree.tools.srs.SRSInfo
```

and you'll get a list of supported CRS in deegree. All CRS that are supported in proj4 are now also supported by deegree.



## Appendix C Featuretype Definition

Path of file: \$wms\_home\$/WEB-INF/conf/wms/featuretypes/

```
<xsd:schema targetNamespace="http://www.deegree.org/app"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:deegreewfs="http://www.deegree.org/wfs"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:app="http://www.deegree.org/app"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd" />
<xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/geometryAggregates.xsd"
/>

<!-- configuration for the persistence backend to be used -->
<!-- This is an automatic generated and manually extended featuretype definition
file; it associates the WFS with the data source(s) -->
<xsd:annotation>
<xsd:appinfo>
<deegreewfs:Prefix>app</deegreewfs:Prefix>
<!--Please refer the demo definitons for accessing HSQLDB, SHAPE or POSTGIS
-->
<deegreewfs:Backend>SHAPE</deegreewfs:Backend>
<!-- This part varies depending on the backend type -->
<deegreewfs:File>../../../../data/utah/vector/SGID500_EnergyResourcesPoly</deegr
eewfs:File>
<!-- This will NOT be set by the creation-tool so please insert the EPSG:
appropriate to your data source SRS -->
<deegreewfs:DefaultSRS>EPSG:26912</deegreewfs:DefaultSRS>
</xsd:appinfo>
</xsd:annotation>
<!-- ===== -->
<!-- here the featuretype name is set, under which the WFS will offer this
featuretype. If you would
like to change this featuretype name make sure, you also change the
type='app:EnergyResourcesType' and
further below the <xsd:complexType name='EnergyResourcesType'> -->
<xsd:element name='EnergyResources' type='app:EnergyResourcesType'
substitutionGroup="gml:_Feature">
<xsd:annotation>
<xsd:appinfo>
<!-- the referenced table in the DB/file is set here. Never change this
unless your sources have changed -->
<deegreewfs:table>SGID500_EnergyResourcesPoly</deegreewfs:table>
<!-- The unique identifier is set here -->
<deegreewfs:gmlId prefix="ID_">
<deegreewfs:MappingField field='ID' type="INTEGER" />
</deegreewfs:gmlId>
</xsd:appinfo>
</xsd:annotation>
</xsd:element>
<!-- ===== -->
<xsd:complexType name='EnergyResourcesType'>
<xsd:complexContent>
<xsd:extension base="gml:AbstractFeatureType">
<xsd:sequence>
<!-- The geometry property will be mapped here. As you can see property
naming (<xsd:element name='geometry') can differ from the source name
(<deegreewfs:MappingField field='GEOM' ...) -->
<xsd:element name='geometry' type='gml:GeometryPropertyType'>
<xsd:annotation>
<xsd:appinfo>
<deegreewfs:Content>
<!-- the srs='' specifies how the data is saved in the
database/file table; for shapes this is -1 as this information is
```

```

 not stored in the file, for databases it depends on the DB-
 Settings, whether the SRS is saved or not. -->
 <deegreewfs:MappingField field='GEOM' type='GEOMETRY' srs='-1' />
 </deegreewfs:Content>
</xsd:appinfo>
</xsd:annotation>
</xsd:element>
<xsd:element name='objectid' type='xsd:integer'>
 <xsd:annotation>
 <xsd:appinfo>
 <deegreewfs:Content>
 <deegreewfs:MappingField field='OBJECTID' type='NUMERIC' />
 </deegreewfs:Content>
 </xsd:appinfo>
 </xsd:annotation>
</xsd:element>
<!-- You are free to change the order of the elements or reduce the number
-->
<!-- By default the featurtypedefinition creation tool sets no
minOccurs=""; minOccurs sets whether or not a property CAN (minOccurs="0")
or MUST (minOccurs="1") be delivered by WFS. If not set minOccurs="1". In
the demos the EnergyResources WMS layer uses this mechanism as it has a
filter condition which requests just some properties of the WFS -->
<xsd:element name='code' type='xsd:string' minOccurs="0">
 <xsd:annotation>
 <xsd:appinfo>
 <deegreewfs:Content>
 <deegreewfs:MappingField field='CODE' type='VARCHAR' />
 </deegreewfs:Content>
 </xsd:appinfo>
 </xsd:annotation>
</xsd:element>
<xsd:element name='shape_area' type='xsd:double' minOccurs="0">
 <xsd:annotation>
 <xsd:appinfo>
 <deegreewfs:Content>
 <deegreewfs:MappingField field='SHAPE_AREA' type='NUMERIC' />
 </deegreewfs:Content>
 </xsd:appinfo>
 </xsd:annotation>
</xsd:element>
<xsd:element name='shape_len' type='xsd:double' minOccurs="0">
 <xsd:annotation>
 <xsd:appinfo>
 <deegreewfs:Content>
 <deegreewfs:MappingField field='SHAPE_LEN' type='NUMERIC' />
 </deegreewfs:Content>
 </xsd:appinfo>
 </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

## Appendix D Deployment Descriptor (web.xml)

Path of file: \$wms\_home\$/WEB-INF/

```
<?xml version="1.0"?>

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
 <display-name>degree 2.2</display-name>
 <description>degree 2.2 OWS</description>
 <servlet>
 <servlet-name>owservice</servlet-name>
 <servlet-class>org.degree.enterprise.servlet.OGCServletController</servlet-
class>

 <init-param>
 <param-name>services</param-name>
 <param-value>wms,wfs</param-value>
 <description>
 list of supported services, e.g.: wfs,wms,wcs (comma separated) allways use
 lowercase
 </description>
 </init-param>

 <!-- WMS INITIALIZING PARAMETERS -->
 <init-param>
 <param-name>wms.handler</param-name>
 <param-value>org.degree.enterprise.servlet.WMSHandler</param-value>
 </init-param>
 <init-param>
 <param-name>wms.config</param-name>
 <param-value>WEB-INF/conf/wms/wms_configuration.xml</param-value>
 </init-param>

 <!-- WFS INITIALIZING PARAMETERS -->
 <init-param>
 <param-name>wfs.handler</param-name>
 <param-value>org.degree.enterprise.servlet.WFSHandler</param-value>
 </init-param>
 <init-param>
 <param-name>wfs.config</param-name>
 <param-value>WEB-INF/conf/wms/LOCALWFS_configuration.xml</param-value>
 </init-param>

 <!-- WCS INITIALIZING PARAMETERS -->
 <!--
 <init-param>
 <param-name>wcs.handler</param-name>
 <param-value>org.degree.enterprise.servlet.WCSHandler</param-value>
 </init-param>
 <init-param>
 <param-name>wcs.config</param-name>
 <param-value>WEB-INF/conf/wms/LOCALWCS_configuration.xml</param-value>
 </init-param>
 -->

 <load-on-startup>1</load-on-startup>
 </servlet>
 <servlet-mapping>
 <servlet-name>owservice</servlet-name>
 <url-pattern>/services</url-pattern>
 </servlet-mapping>
 <welcome-file-list>
 <welcome-file>/index.jsp</welcome-file>
 </welcome-file-list>
 <error-page>
 <error-code>500</error-code>
 <location>/error.jsp</location>
 </error-page>
</web-app>
```

```
</error-page>
<error-page>
 <exception-type>org.deegree.ogcwebservices.OGCWebServiceException</exception-
 type>
 <location>/error.jsp</location>
</error-page>
</web-app>
```