

OGC 06-045r1

Open Geospatial Consortium Inc.

Date: 2005-07-24

Reference number of the OpenGIS® Project Document: **OGC 06-045r1**

Version 0.9

Category: Discussion Paper

Editor: Eric LaMar

OGC Web Map Service – Proposed Animation Service Extension

**Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.**

Warning

This document is not an OGC Standard nor is it an official position of the OGC or the OGC membership. It is distributed for review and comment. The document is subject to change without notice and may not be referred to as an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OpenGIS® Discussion Paper
Document stage: Proposed Revision to an OGC Specification
Document language: English

Acknowledgements: Special Thanks to Scott Hamilton, Galaxy Global, for his hard work and diligence in writing this document.

Table of Contents

i. Preface.....	iv
ii. Document terms and definitions	iv
iii. Submitting Organizations	iv
iv. Document Contributor Contact Points.....	iv
v. Revision History	v
vi. Changes to the OpenGIS® Abstract Specification	v
v. Future work	v
Foreword	vi
0. Introduction.....	vi
0.1 Issues Addressed by ASE	vii
0.2 Note of Interest.....	vii
1. Scope.....	1
2. Compliance	1
3. Normative references	1
4. Terms and definitions.....	2
5. Conventions	3
5.1 Abbreviated terms	3
6. Web-Map-Server Overview	3
6.1 A review of WMS 1.3	3
6.2 General HTTP Request Rules used by WMS and ASE.....	5
7.0 Animation Layers.....	6
7.1 AES Introduction Material	6
7.2 Enhancements to WMS GetCapabilities for ASE.....	9
8.0 ASE Multiple Frame Definitions	10
8.1 Description of ASE MFD extension	10
8.1.1 Pros to the ASE MFD extension.....	10
8.1.2 Cons to the ASE MFD extension.	10
8.1.3 Proof of Concept ASE MFD Implementation	10
8.2 ASE MFD Root Element.....	11
8.3 ASE MFD Returned Layer Descriptors	11
8.4 Example WMS Requests using an ASE MFD (GetFrames).....	12
8.4.1 General.....	12
8.4.2 GetFrames request overview	12
8.4.3 Request Parameters.....	13
9.0 ASE Animation File Definitions.....	15
9.1 Description of ASE AFD extension	15
9.1.1 Pros to the ASE AFD extension	15

9.1.2 Cons to the ASE AFD extension	15
9.1.3 Proof of Concept ASE AFD Implementation.....	15
9.2 ASE AFD Root Element	17
9.3 ASE AFD Returned Layer Descriptors	17
9.4 Example WMS Requests using an ASE AFD (GetAnimation)	18
9.4.1 General.....	18
9.4.2 GetAnimation request overview	18
9.4.3 Request Parameters.....	19
Annex A Compliance Testing.....	21
Annex B: Animation Service Extension Schema	23

i. Preface

This document explains how the Web Map Server (WMS 1.0 [1] & 1.1 [2,3]) specification can be extended to allow animations that move in space over time. It should be read in conjunction with the latest version WMS specification. When this document was written, the latest version WMS specification was defined by the WMS 1.1.1 Specification.

Suggested additions, changes, and comments to this Discussion Paper are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, turn on change tracking, and then make your suggested changes.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Submitting Organizations

This Discussion Paper was submitted to the OGC by the following organizations:

Institute for Scientific Research, Inc.

Galaxy Global Corp.

iv. Document Contributor Contact Points

All questions regarding this Discussion Papershould be directed to the editor or the contributors:

<i>COMPANY</i>	<i>CONTACT</i>	<i>EMAIL</i>
Galaxy Global Corp.	Scott Hamilton	shamilton@galaxyglobal.com
Institute for Scientific Research, Inc.	Eric LaMar	elamar@isr.us
Institute for Scientific Research, Inc.	Ryan Boyles	rboyles@isr.us
Institute for Scientific Research, Inc.	Fred Bernal	fbernal@isr.us
Institute for Scientific Research, Inc.	Matt McMahan	mmcmahon@isr.us

v. Revision History

<i>Date</i>	<i>Release</i>	<i>Author</i>	<i>Paragraph modified</i>	<i>Description</i>
7/24/06	0.9	Carl Reed	Numerous	Get ready for public publication

vi. Changes to the OpenGIS® Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

v. Future work

Improvements in this document are desirable to further describe the necessary data structures to implement the specification.

Foreword

This document includes two annexes; both Annexes are normative. Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

0. Introduction

WMS is a web-based tool that produces and serves digital maps. In the past, analyzing and creating maps was a slow and labor-intensive process. Now with Global Positioning Systems and other digital methods of gathering geographical data, digital mapping has become an important component in several fields. The WMS specification is a web-based protocol that allows communication of map-style data between a server and a client.

WMS, and digital imaging in general, has a wide range of applications. There are several practical uses of WMS, such as weather forecasting, tourism, and transportation. Google, for example, has a browser called Google Earth that accesses several WMS servers. It is a 3-D browser that contains layers including restaurants, road maps, recreational locations, landmarks, as well as satellite imagery. Their browser is proprietary and prevents you from adding your own WMS server to its list, but it is a perfect example of the power of WMS.

Work began with WMS through installing the University of Minnesota's custom implementation of a WMS server called MapServer. UMN packaged MapServer into a Linux-based operating system called HOST GIS (Global Information Systems). HOST GIS houses all of the code (written in C and C++) that UMN wrote to implement their map server. UMN MapServer has two major components it utilizes in order to produce a map. The first is called a map file. A map file contains information on multiple layers of images, and a "bounding box" of latitude and longitude coordinates for the images. The second, an ArcView World File (WLD), is a file that is written for each and every image on the UMN map server. The WLD file contains specific latitude and longitude coordinates, as well as the pixel size of the images. These two types of files have to work hand-in-hand to make images appear properly.

However, UMN's MapServer is not fully compliant with the WMS specification, which means that in order to do animations, one must not use UMN's implementation but rather write an implementation of map server or purchase an alternative. One major issue that was encountered is that UMN did not implement the time extents in a robust manner, limiting the ability to change a map image over time. Obviously, this is a necessity in implementing any kind of animation.

As a result, an implementation of WMS was developed at the Institute for Scientific Research to address the issues with UMN's implementation. ISR's implementation has many limitations and does not implement the entire specification. Only those pieces necessary to demonstrate the ability of the specification to serve animated maps, which transverse both time and space, was implemented. Moving animations were very successfully served to NASA's World Wind (<http://worldwind.arc.nasa.gov/>) WMS client. To the knowledge of the team, World Wind is the only freely-available WMS-enabled 3-D client to support animations.

0.1 Issues Addressed by ASE

We found a few major problems in the WMS Specification with regard to animations. Therefore, we are proposing an extension to the specification to address these issues.

1. Using a fully compliant WMS server, it is not possible to specify an animation whose Bounding Box scales and moves over time, as the entire layer shares the bounding box.
2. Specify a bounding box only for an animation that would contain the entire life cycle of the animation.
3. The present specification would not allow for a client to determine where valid data is contained in a specific frame -- making links to Web Coverage Service Data from within an animation sequence difficult. The ability to shift the bounding box gives the client exact valid bounds for an animation frame -- making links to Web Coverage Service Data possible.
4. Re-sampling errors occur when the server attempts to move an image by placing a scaled version of the image into a larger bounding box.
5. Animations of weather events, such as hurricanes suffer in accuracy of location due to the opaque box around image formats that do not support transparency.

0.2 Note of Interest

It must be noted that the example XML data, URLs, and results shown in the Proof of Concept Sections do not follow the specification. They are included to show that implementation is possible and as a measure of the benefits. The Proof of Concept code was written prior to the detailed specification extension and therefore does not follow the specification in detail, only in concept. The Proof of Concept sections are not meant to be an example implementation.

1. Scope

This OpenGIS® Interoperability-Program Report specifies the format of an Animation Service Extension for producing georeferenced animations. Different modes for utilizing this animation service extension specification are discussed.

2. Compliance

Compliance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3. Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO 19105:2000, *Geographic information — Conformance and Testing*

OGC 05-008, *OpenGIS® Web Services Common Specification*

CGI, *The Common Gateway Interface*, National Center for Supercomputing Applications,
<<http://hoohoo.ncsa.uiuc.edu/cgi/>>

IETF RFC 2045 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Freed, N. and Borenstein N., eds., <<http://www.ietf.org/rfc/rfc2045.txt>>

IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol -- HTTP/1.1*, Gettys, J., Mogul, J., Frystyk, H. Masinter, L., Leach, P., and Berners-Lee, T., eds., <<http://www.ietf.org/rfc/rfc2616.txt>>

IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee, T., Fielding, N., and Masinter, L., eds., <<http://www.ietf.org/rfc/rfc2396.txt>>

OGC Abstract Specification 12 (January 2002), *The OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture (Version 4.3)*, Percivall, G. (ed.),
<<http://www.opengis.org/techno/abstract/o2-112.pdf>>

OGC Adopted Implementation Specification: Web Map Service version 1.3, August 2004, OGC document OGC 04-024

OGC Adopted Implementation Specification: Web Map Server version 1.1.1, February 2002, OGC document OGC 01-068r2

OGC Adopted Implementation Specification: Web Map Service version 1.1.0, December 2001, OGC document OGC 01-047r2

OGC Adopted Implementation Specification: Web Coverage Service (WCS), Version 1.0.0, August 2003, document OGC 03-065r6

XML 1.0 (February 2004), *Extensible Markup Language (XML) 1.0 (3rd edition)*, World Wide Web

Consortium Recommendation, Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., and Yergeau, F., eds., <<http://www.w3.org/TR/2004/REC-xml-20040204> >

This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

4. Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] shall apply. In addition, the following terms and definitions apply.

4.1

operation

Specification of a transformancy or query that an object may be called to execute [OGC AS 12]

4.2

interface

Named set of **operations** that characterize the behavior of an entity [OGC AS 12]

4.3

service

Distinct part of the functionality that is provided by an entity through **interfaces** [OGC AS 12]

4.4

service instance or server

Actual implementation of a **service**

4.5

client

Software component that can invoke an **operation** from a **server**

4.6

request

Invocation of an **operation** by a **client**

4.7

response

Result of an **operation** returned from a **server** to a **client**

4.8

map

Pictorial representation of geographic data

4.9

capabilities XML

Service-level metadata describing the **operations** and content available for a given **service instance**

4.10

animation

A series of **maps**, or georeference image data along with specified time stamps and/or frame-rates to represent a change over time.

5. Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

AFD	Animation File Definition
ASE	Animation Services Extension
CGI	Common Gateway Interface
CRS	Coordinate Reference Systems
DCP	Distributed Computing Platform
DTD	Document Type Definition
EPSG	European Petroleum Survey Group
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extensions
MFD	Multiple Frame Definition
OGC	Open GIS Consortium
OWS	OGC Web Service
PNG	Portable Network Graphics
RFC	Request for Comments
SFD	Single Frame Definition
SLD	Styled Layer Descriptor
SRS	Spatial Reference System
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
WebCGM	Web Computer Graphics Metafile
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
XML	Extensible Markup Language

6. Web-Map-Server Overview

6.1 A review of WMS 1.3

The WMS Specification consists of two basic system calls. The first is **GetCapabilities** which returns an XML document adhering to a standard set of parameters. This XML document gives details about

map layers, the server's capabilities (i.e. the image formats, projections, and geographic bounds of the server.) See Appendix A for a sample capabilities XML document with additional comments. Below is a sample **GetCapabilities** request:

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?request=GetCapabilities&service=WMS&version=1.1.1>

The second is **GetMap** which defines the format for requesting a map (image) from a WMS server based on the capabilities file. The best way to define the **GetMap** request is to give an example request and break down the details.

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Charley&format=image/png&width=512&height=512&time=2004-08-12T15:55Z&srs=EPSG:4326&bbox=-91.00,11.8,-68,34.50&styles=&transparent=TRUE>

Every WMS request has certain components regardless of whether a map or a capabilities document is requested. See table below.

Server Name	<i>http://hostgis.isr.us</i>
Script Path	<i>/cgi-bin/scottwms/wms.pl</i>
Service Name	<i>Service=WMS</i> (OGC defines other services i.e. WFS, etc.)
Version	<i>Version=1.1.1</i> (Versions are not entirely backward compatible so the version number is required to determine the proper format for the request.)
Request Type	<i>request=GetMap</i> or <i>request=GetCapabilities</i> (in Version 1.1.1)

The parameters to be used for creating animations come from the **GetMap** request type and are outlined below.

Layer Name	<i>layer=Charley</i> (Any layer from the XML capabilities file)
Image format	<i>format=image/png</i> (Any of several formats, the sever's capabilities file will list all formats a server is capable of delivering. Our sample server is limited to png.)
Image width	<i>width=512</i> (can be any size -- the output image will be this number of pixels wide).
Image height	<i>height=512</i> (can be any size -- the output image will be this number of pixels high).
Image time (Optional needed for animations, not required for a compliant sever implementation.)	<i>time=2004-08-12T15:55Z</i> (the time stamp of a particular image, time is one of many user defined constraints that can be placed on a map image and usually ties to a database field that contains the image for a particular time, these custom constraints are defined as a part of the capabilities document, and, according to the specification, must be optional in order to be compliant.
Image Projection	<i>srs=EPSG:4326</i> (This can be any EPSG Projection code. The WMS specification requires all compliant map servers to provide imagery in

Layer Name	<i>layer=Charley</i> (Any layer from the XML capabilities file)
	at least EPSG:4326 so that was chosen for our example). SRS is the Spatial Reference System used by the WMS Server for this particular layer.
Geographic Bounding Box	<i>bbox=-91.00,11.8,-68,34.50</i> (The geographical coordinates of the corners of the image upper-left, lower-right specified in the Projection model from the SRS field.) This is the total area in which the client wants the data projected into. It can not be larger than the bounding box specified by the capabilities document, but it can be smaller, in which case only a portion of the defined image will be displayed.
Output styles	<i>styles=</i> Optionally specify a format for the output document.
Transparency of the image	<i>transparent=TRUE</i> (Specifies whether the output image should contain transparency information.)

There is one other aspect of the specification that provides animations that move geographically over time. The specification serves georeferenced images. The Geospatial Data Abstraction Library (GDAL) is a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license. As a library, it presents a single abstract data model to the calling application for all supported formats. The related OpenGIS Simple Features Reference Implementation (OGR) library (which resides within the GDAL source tree) provides a similar capability for simple features vector data. GDAL and associated libraries can be found at: <http://www.gdal.org>.

The methods within GDAL allows exact specification of coordinates for an image, and therefore, the ability to place the image for a specified time, within its own bounding box nested within the bounding box requested by the client. This results in an image the size of the requested bounding box being generated with an opaque, or transparent background depending on the image format. GDAL is one of many available libraries for Geospatial data. It does not solve the shortcomings of the WMS specification with regards to animations.

6.2 General HTTP Request Rules used by WMS and ASE

At present, the only distributed computing platform (DCP) explicitly supported by OGC Web Services is the World Wide Web itself, or more specifically Internet hosts implementing the Hypertext Transfer Protocol (HTTP) [IETF RFC 2616]. Thus the On-line Resource of each operation supported by a service instance is an HTTP Uniform Resource Locator (URL). At the discretion of the service provider the URL may be different or the same for each operation. Each URL **shall** conform to the description in [IETF RFC 2616] (Section 3.2.2 "HTTP URL"), but is otherwise implementation-dependent; only the parameters comprising the service request itself are mandated by the OGC Web Services specifications.

HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular OGC Web Service type and offered by a service instance, and the use of the Online Resource URL differs in each case. The basic WMS specification defines both HTTP GET and HTTP POST for invoking operations. (The WMS Animation Service Extension only defines HTTP GET for all operations.)

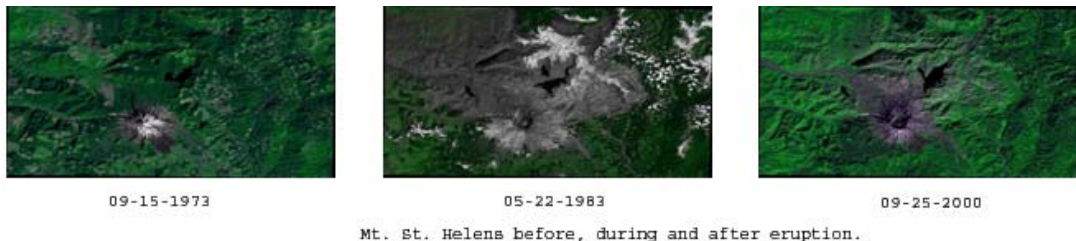
7.0 Animation Layers

7.1 AES Introduction Material

Before going into a detailed breakdown of how to implement moving animations, let us talk a little about how animations work in general, and then discuss the subtle changes that must occur to move the animation over a geographical path with time.

The first experimentation with animations began by exploring the NASA World Wind Project. World Wind is a client containing custom scripts to provide pseudo-animations by displaying a series of still imagery. In essence, the WMS server does not send animations to World Wind, or any other client for that matter, but sends a series of time-stamped imagery. The WMS specification does not provide a method of serving animations, as its primary focus is mapping with still imagery; however, it does provide a key element necessary for animations: the ability to associate a time with an image. The specification also allows for individual images within a map layer to have various georeferenced data, allowing the pseudo-animation to move geographically over time.

First take a look at a simple animation-- for example, a volcano before, during, and after an eruption. Images obtained from NASA's Scientific Visualization Server (<http://svs.gsfc.nasa.gov>).



In order to animate this event, one must place the images on the server and provide a geographic bounding box and a series of time stamps within a layer. This will give an XML capabilities document with the follow layer item within it.

```
<Layer>
  <Name>Mt_St_Helens</Name>
  <Title>Mount St. Helens Before, During, and After (1024x1024 Animation)</Title>
  <Abstract>Mount St. Helens erupted on May 18, 1980, devastating more than 150 square miles of forest in southwestern Washington state. This animation shows Landsat images of the Mount St. Helens area in 1973, 1983, and 2000, illustrating the destruction and regrowth of the forest. The 1983 image clearly shows the new crater on the northern slope where the eruption occurred, the rivers and lakes covered with ash, and the regions of deforestation. The 2000 image, taken twenty years after the eruption, still shows the changed crater, but much of the devastated area is covered by new vegetation growth.</Abstract>
  <LatLonBoundingBox minx="-122.36011" miny="46.122115" maxx="-121.97916" maxy="46.384547" />
  <BoundingBox SRS="EPSG:4326" minx="-122.36011" miny="46.122115" maxx="-121.97916" maxy="46.384547" />
  <Dimension name="time" units="ISO8601"/>
  <Extent name="time" default="09-25-2000">09-15-1973,05-22-1983,09-25-2000</Extent>
</Layer>
```

The client will then know that the following three map requests will produce the three frames of the animation.

http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Mt_St_Helens&format=image/png&width=512

http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Mt_St_Helens&format=image/png&width=512&height=512&time=09-15-1973&srs=EPSG:4326&bbox=-122.36011,46.122115,-121.97916,46.384547&styles=&transparent=TRUE

http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Mt_St_Helens&format=image/png&width=512&height=512&time=05-22-1983&srs=EPSG:4326&bbox=-122.36011,46.122115,-121.97916,46.384547&styles=&transparent=TRUE

http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Mt_St_Helens&format=image/png&width=512&height=512&time=09-25-2000&srs=EPSG:4326&bbox=-122.36011,46.122115,-121.97916,46.384547&styles=&transparent=TRUE

It is now up to the client application to play the frames in sequence and, in effect, create the animation. As one can see, this is a pseudo-animation. In fact, WMS does not actually send an animation to the client, but rather a series of still images. The result, of course, will be an animation.

Imagery of a hurricane (three frames over a period of three days) is an example of a simple moving animation. *Images obtained from NASA's Scientific Visualization Server (<http://svs.gsfc.nasa.gov>).*



08-11-2004



08-12-2004



08-13-2004

Hurricane Charley Imagery

In order to animate this event, one must place the images on the server and provide a geographic bounding box for each image, and a series of time stamps within a layer. The server will then re-sample the images creating an image that covers the complete bounds of the animation. This is done by mapping the original image over the larger transparent image. However this creates a limitation in that only Portable Network Graphics (png), Graphic Interchange Format (gif), or Tagged Image File Format (TIFF) images, which support transparency, may be used without causing an opaque black background on the animation. This does not cause a server to fall out of specification, but does create issues for some client software.

In order to move the animation over time, one must set the geographical coordinates of the imagery. This is done using a World coordinates (WLD) file to georectify the imagery. This could be done in any number of ways including a database field. However, this implementation uses the WLD files

specified within the GDAL library. The table below shows the WLD files used for each image above to correctly locate them along with a description of each field in the file.

<i>Meaning of Number</i>	<i>2004-08-11.wld</i>	<i>2004-08-12.wld</i>	<i>2004-08-13.wld</i>
x scaling factor	0.014013	0.01542	0.01539
x rotation term.	0.00	0.00	0.00
Y rotation term.	0.00	0.00	0.00
Y scaling factor	-0.01054	-0.01142	-0.01626
Upper Left Longitude	-83.166	-88.8844	-90.515327
Upper Left Latitude	23.39	26.8436	34.26223

This will give an XML capabilities document with the following layer item within it.

```
<Layer>
  <Name>Charley</Name>
  <Title>Hurricane Charley Animation</Title>
  <LatLonBoundingBox minx="-90.5253" miny="12.6028" maxx="-68.8210" maxy="34.2722" />
  <BoundingBox SRS="EPSG:4326" minx="-90.5253" miny="12.6028" maxx="-68.8210" maxy="34.2722" />
  <Dimension name="time" units="ISO8601"/>
  <Extent name="time" default="2004-08-11Z" nearestValue="0">2004-08-11T18:15Z,2004-08-12T15:55Z,2004-08-13T16:35Z</Extent>
</Layer>
```

The client will then know that the following three map requests will produce the three frames of the animation.

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Charley&format=image/png&width=512&height=512&time=2004-08-11T18:15Z&srs=EPSG:4326&bbox=-90.5253,12.6028,-68.8210,34.2722&styles=&transparent=TRUE>

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Charley&format=image/png&width=512&height=512&time=2004-08-12T15:55Z&srs=EPSG:4326&bbox=-90.5253,12.6028,-68.8210,34.2722&styles=&transparent=TRUE>

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetMap&layers=Charley&format=image/png&width=512&height=512&time=2004-08-13T16:35Z&srs=EPSG:4326&bbox=-90.5253,12.6028,-68.8210,34.2722&styles=&transparent=TRUE>

Notice that the requests are almost identical to the requests for the still imagery with the exception of the layer name. This is because the movement of the imagery is done on the server side and the client always receives an image that fills the entire geographical bounding box from the request. The image will have transparent edges around the true image to locate the image in the correct geographical location. This transparent image is generated by the server at the time of the request. The first three images below will be returned from the server. The fourth overlaid image indicates the motion in the animation.

Once again, the client application plays the frames in sequence and, in effect, creates the animation. It is clear from this discussion that an extension to the specification for animations would greatly benefit those people working with time sensitive data such as weather forecasting and flight tracking.

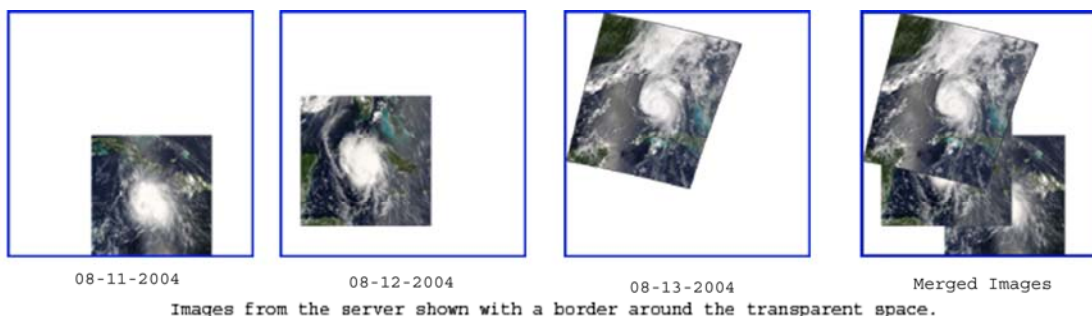
One major issue with this method of a moving animation is that it requires the imagery to be re-sampled to create the animation frame, nearly always degrading the quality of the data. The lack of a method for serving moving imagery without degrading the data quality indicates a great need for extending the specification.

From research it appears there are two general ways in which the specification could be extended to provide animations more effectively. Either one or both methods may be used to implement a compliant ASE capable WMS Server: (8.0) Multiple Frame Definition (9.0) Animation File Definition.

7.2 Enhancements to WMS GetCapabilities for ASE

To insure valid interaction with a range of WMSes, including integrated and component WMS, the capabilities response has been enhanced. Specifically, it can describe the following information:

1. Does the WMS provide ASE support?



2. Does the WMS support Multiple Frame Requests **GetFrames**?
3. Does the WMS support Animation File Requests **GetAnimation**?

The fact that a WMS server provides ASE support is indicated by the presence of any one of the request types appearing in the capabilities document. The last two items that need to be described using the capabilities are handled via the **Request** element of WMS capabilities, with the addition of any one or both of the new request types represented by the **GetFrames** and **GetAnimation** elements. The **Request** element can be used to describe whether a request can be handled by HTTP (as a particular instance of a distributed computing platform) and, if so, whether the request can use HTTP GET and/or POST.

```
<!-- ASE Interfaces: Presence of any element listed below means this server support ASE in some capacity. -->
<!ELEMENT GetFrames (Format+, DCPTType+)><!-- Support for ASE MFD-->
<!ELEMENT GetAnimation (Format+, DCPTType+)><!-- Support for ASE AFD-->
```


8.0 ASE Multiple Frame Definitions

8.1 Description of ASE MFD extension

Instead of returning a single image, GetFrames would return a series of images with different bounding box parameters. This would be an extension of the frame type specified in section 5.1. This would allow the client to properly locate the returned set of imagery and playback an animation with a single request.

8.1.1 Pros to the ASE MFD extension.

This model includes the pros from the model discussed in section 5.1. In addition to these pros, this model addresses issues with high-latency networks. The ability to request multiple frames with a single request limits the communications from client to server, decreasing the network latency because the bulk of traffic will be from server to client.

A big advantage to this model is that it does not limit the ability of the client to request only certain frames of an animation. This could allow the client to download any portion of an animation for viewing.

The animation will be returned to the client as a single stream tremendously simplifying the communications model over the current model and model 5.1.

Returned as a multi-part MIME type, the resulting stream can be viewed in most modern web browsers allowing users without specific client software to view the frames of the animation.

8.1.2 Cons to the ASE MFD extension.

This extension model has the same disadvantages as the previous model with the additional disadvantage of creating a very complex MIME return type resulting in a more complicated client side interpretation of the data. This implementation would require a higher performance client machine.

8.1.3 Proof of Concept ASE MFD Implementation

An implementation of the model was written in Perl extending a standard WMS implementation. A GetFrames request type was implemented which takes the same parameters as GetMap, except that it returns a sequence of XML bounding box parameters followed by an image file, followed by the next frame in the same format. This model implementation may not be fully compliant with the XML standards, but close enough for proof of concept. The returned document looks like the following:

```
Content-Type: multipart/mixed; boundary=ISRSCOTTWMSVER01010101
```

```
--ISRSCOTTWMSVER01010101
```

```
Content-Type: text/xml; charset=ISO-8859-1
```

```
<BoundingBox SRS="EPSG:4326" minx=" 32.4012600 " miny=" -89.5012600" maxx=" 29.8207800 " maxy=" -87.0871000" />
```

```
Content-Type: image/png
```

```
%PNG (The binary image data)
```

```
--ISRSCOTTWMSVER01010101
```

```
Content-Type: text/xml; charset=ISO-8859-1
```

```
<BoundingBox SRS="EPSG:4326" minx=" 32.4012600 " miny=" -89.5012600" maxx=" 29.8207800 " maxy=" -87.0871000" />
```

```
Content-Type: image/png
```

```
%PNG (The binary image data)
```

```
//sequence repeats until all frames are served.
```

```
--ISRSCOTTWMSVER01010101--
```

The client application must then split the multi-part MIME into many parts: the XML documents and image files. In the proof of concept code, a modified NASA World Wind client was used. The XML portion of the data was processed in memory and the raw image files were stored for processing. Each image/XML file pair was treated as if it were a separate download request in the client proving that the streaming method works, without having to recode a lot of the display engine.

The proof of concept code showed that this model worked very well under most circumstances. It allowed the server to send multiple images and display them properly located regardless of the requested bounding box. The proof of concept code needs to be extended to crop images that fall on the edges of the requested bounding box before it can be considered a complete implementation.

In this manner there was substantial performance improvement over both the original implementation and ASE SFD at the server. However, the animations appear slower on the client due to the rendering model within World Wind. It will require some extra work on the client side to see the same level of performance as with server-side image processing.

It was discovered that the data size increase returning multiple frames in a single stream was substantial and created a significant delay before the client could display the first frame. However, subsequent frames displayed more rapidly and network traffic was reduced.

Before moving on to implement ASE AFD, some tests were run by requesting a large 240 frame animation of Hurricane Katrina's landfall event. The high resolution imagery and sheer size of the animation created major problems for both the client and the server due to the size of the data returned. The animation still took around 15 minutes to download, and in that time frame, the client is waiting for the first frame, proving the need for additional compression of multi-frame animations.

It was discovered that the client performance improvement was negligible over ASE MFD. From a computer standpoint ASE MFD is slightly faster, but as an end user, the server-side model appears faster since there is less time between a request and viewing the first frame of the animation.

In conclusion, when it comes to high-resolution, multiple frame animations, neither suggested model server-side or ASE MFD is very effective, and leaves the user waiting for imagery to transfer from the server. However, they serve the purpose of allowing an animations bounding box to move and scale over time.

8.2 ASE MFD Root Element

An ASE MFD document is defined as a sequence of time stamped layers. The root element is defined by the WMS layer specification. There is no modification to the layer definitions required.

8.3 ASE MFD Returned Layer Descriptors

An ASE MFD request will return a multipart/mixed mime type document. The first part of the

document is an XML document based on the Layer class of the WMSCapabilities document type. This will be followed by a mime-encoded image of the requested format. The pattern will then repeat until all requested frames have been received.

Below is an example of a returned XML Document.

```
<?xml version="1.0" encoding="UTF-8"?>
<Layer xmlns="http://www.opengis.net/ase" schemaLocation="http://www.opengis.net/ase/capabilites.xsd">
<!-- First Frame Definition -->
<Layer>
<Dimension name="time" units="ISO8601"/>
<Extent name="time">1995-08-03</Extent>
<BoundingBox SRS="EPSG:4326" minx="-91" miny="25.24" maxx="-79.36" maxyy="32.20"/>
</Layer>
```

8.4 Example WMS Requests using an ASE MFD (GetFrames)

8.4.1 General

The GetFrames operation returns a single frame of an animation. In all respects, it is a map with timestamped geographically rectified image data. Upon receiving a GetFrame request, a WMS with ASE SFD shall either satisfy the request or issue a service exception.

8.4.2 GetFrames request overview

The following table describes the GetFrame Request.

<i>Request Parameter</i>	<i>Mandatory/Optional</i>	<i>Description</i>
VERSION=1.3.0	Mandatory	Request version.
REQUEST=GetFrames	Mandatory	Request name.
LAYER=layer	Mandatory	A single map layer containing an animation sequence of frames. Determined by the presence of a Dimension called "time".
STYLE=style_list	Mandatory	The rendering style requested for the animation frame.
CRS=namespace:identifier	Optional (Default CRS of the server will be returned as part of the XML document.)	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	Optional (Default Bounding Box of the frame will be returned as part of the XML document.)	Bounding box corners (lower left, upper right) in CRS units
WIDTH=output_width	Optional (The default image width will be returned.)	Width in pixels of map picture.
HEIGHT=output_height	Optional (The default image height will be returned.)	Height in pixels of the map picture.
FORMAT=output_format	Mandatory	Output format of the map.

<i>Request Parameter</i>	<i>Mandatory/Optional</i>	<i>Description</i>
TRANSPARENT=TRUE/FALSE	Optional	Background transparency of the map (default=TRUE)
BGCOLOR=color_value	Optional	Hexadecimal red-green-blue color value for the background color (default=0xFFFFFF)
EXCEPTIONS=exception_format	Optional	The format in which exceptions are to be reported by the WMS (default=XML)
TIME=time	Mandatory	Time values of the requested frames.
ELEVATION=elevation	Optional	Elevation of the layer/frame requested.
Other sample dimension(s)	Optional	Value of other dimensions as appropriate.

8.4.3 Request Parameters

8.4.3.1 VERSION

The mandatory VERSION parameter is defined in 6.9.1 of the OpenGIS Web Map Service Implementation Specification version 1.3 OGC Document number OGC 04-024. The value “1.3.0” shall be used for GetFrames requests that comply with this International Standard.

8.4.3.2 REQUEST

The mandatory REQUEST parameter is defined in 6.9.2 of the OpenGIS Web Map Service Implementation Specification version 1.3 OGC Document number OGC 04-024. To invoke the GetFrames operation, the value “GetFrames” shall be used.

8.4.3.3 LAYER

The mandatory LAYER parameter is the map layer to be returned by the GetFrames request. The value of the LAYER parameter is a valid layer name. Allowed layer names are the character data content of any <Layer><Name> element in the service metadata that has an associated Dimension called time.

8.4.3.4 STYLE

The mandatory STYLE parameter lists the style in which the frame is to be rendered. The value of the STYLE parameter is a valid style name. The style name shall be one that was defined in a <Style><Name> element that is either directly contained within, or inherited by, the associated <Layer> element in service metadata. (In other words, the Client may not request a Frame in a Style that was only defined for a different Layer.) A server shall throw a service exception (code = StyleNotDefined) if an unadvertised Style is requested. A client may request the default Style using a null value (as in STYLE=).

If the server advertises several styles for a layer, and the client sends a request for the default style, the choice of which style to use as default is at the discretion of the server. The ordering of styles in the

service metadata does not indicate which is the default.

8.4.3.5 CRS

The CRS request parameter states which Layer CRS (Section 6.7.3 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024) applies to the BBOX request parameter. The value of the CRS parameter in a request to a particular server shall be one to the values defined in that server's service metadata in a <CRS> element defined or inherited by the requested layer.

For more information on the CRS parameter refer to section 7.3.3.5 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

8.4.3.6 BBOX

The optional BBOX parameter allows a Client to request a particular Bounding Box or region of a map layer. The details of the BBOX parameter are outlined in section 7.3.3.6 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

8.4.3.7 FORMAT

The mandatory FORMAT parameter states the desired format of the frame/map. Support values for a GetFrame request on a WMS server are listed in one or more <Request><GetFrame><Format> elements of its service metadata. The entire MIME type string in <Format> is used as the value of the FORMAT parameter. There is no default format, however the FORMAT parameter only affects the embedded image format. The return type for frames is always *multipart/mixed* as the frames contain default bounding box and time parameters in a preceding xml document for each frame element. In an HTTP environment, the MIME type shall be set on the returned object using the Content-type entity header, and it shall be *multipart/mixed*. The requested FORMAT MIME type will be embedded in the multipart/mixed data as a subtype using the Content-type entity header. If the request specifies a format not supported by the server, the server shall issue a service exception (code=InvalidFormat).

8.4.3.8 WIDTH, HEIGHT

The optional WIDTH and HEIGHT parameters specify the size in integer pixels of the frame to be produced. The details of the parameters are outlined in section 7.3.3.8 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

8.4.3.9 TIME

The mandatory TIME parameter is defined as a time value to be expressed as defined in Section 6.7.5 and Annexes C and D of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024. For the GetFrames request, the time parameter must result in multiple time stamped images to be returned.

8.4.3.10 Other Optional Parameters

The additional optional parameters are clearly defined in the subsections of section 7.3 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

The options are used in exactly the same manner as they are for GetMap requests.

9.0 ASE Animation File Definitions

9.1 Description of ASE AFD extension

Instead of returning an image, or series of images, GetAnimation would return an avi, animated gif, or other movie format along with an XML document defining the time stamp, and georeferencing data locating each frame to allow the client to move the animation along the path. These would be embedded in a multi-part MIME type.

9.1.1 Pros to the ASE AFD extension

The returned data set is much smaller than returning multiple frame images. Movies are much smaller in size due to the nature of video compression algorithms. Each frame, rather than being stored as a full image, is stored as a variance from the prior image. This compression would be very effective in hurricane models as the change is minor between individual frames over time giving the ability to serve higher resolution imagery at a much higher rate. Streaming the movie would allow for an apparently faster download of the animation to the client.

9.1.2 Cons to the ASE AFD extension

Individual frames or a subset of the frames would not be readily available to the client unless the server generated the movie on the fly. Generating the movie on the fly would create large processor overhead on the server, as well as create a delay in rendering the movie.

There is not a movie format available aside from animated gif that will allow for transparency. This prevents the client from easily overlaying the movie onto a map or other image and comparing the movie to the underlying imagery. This can be overcome by extracting the frames from the movie as individual images with transparency settings on the client side.

9.1.3 Proof of Concept ASE AFD Implementation

An implementation of the model was written in Perl extending a standard WMS implementation. A GetAnimation request type was implemented which takes the same parameters as GetMap, excluding the bounding box parameters as they are returned from the server. The GetAnimation request in our proof of concept code returns a multi-part MIME type. The returned data includes an XML document that contains the bounding box and time stamp parameters for each frame of the animation. The XML document is followed by an animation sequence of images that are compressed using MSMPEG4 encoding. In our implementation the animation sequence is encoded by the server and can only be returned to the client as a whole. However, there are tools available to allow individual frames of the animation to be sent from the server upon request, as well as the ability to encode the frames on the fly at the server side if so desired.

The implementation proved that both of the Cons to the implementation model can be easily overcome. The inability to serve individual frames is overcome by software tools that allow transmission of individual frames of a compressed video file over a stream, making a subset of the frames available to the client without the need to generate the movie on the fly at the server level. The inability to serve

transparent images is overcome by setting a particular image color to transparent at the client level in the animation and using tools of the client to convert the compressed animation into individual frames using any image format the client desires.

Under the hood, this implementation works by encoding the imagery as a MSMPEG4 movie. This movie is appended to a multi-part MIME document containing information about each frame. This is sent to the client. Any web browser can receive the multi-part MIME and has the capabilities of splitting the XML from the movie and playing the movie. However, the standard web browser does not know how to handle the georeferencing information so the animation will not move in space.

The implementation when given the url:

<http://hostgis.isr.us/cgi-bin/scottwms/wms.pl?service=WMS&version=1.1.1&request=GetAnimation&layers=ErinTrack>

returns a Multipart MIME type document similar to the following:

Content-Type: multipart/mixed; boundary=ISRSCOTTWMSVER01010101

--ISRSCOTTWMSVER01010101

Content-Type: text/xml; charset=ISO-8859-1

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://schemas.opengespatial/wms/1.1.1/WMS_MS_Capabilities.dtd">
<LAYER>
<BoundingBox SRS="epsg:4326" minx="21.3" miny="-74.2" maxx="23.3" maxy="-72.2" time="1995-07-31T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="21.6" miny="-74.6" maxx="23.6" maxy="-72.6" time="1995-07-31T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="21.8" miny="-74.9" maxx="23.8" maxy="-72.9" time="1995-07-31T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="22.2" miny="-75.3" maxx="24.2" maxy="-73.3" time="1995-07-31T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="22.6" miny="-75.9" maxx="24.6" maxy="-73.9" time="1995-08-01T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="23.3" miny="-76.7" maxx="26.3" maxy="-74.7" time="1995-08-01T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="24.5" miny="-77.3" maxx="26.5" maxy="-75.3" time="1995-08-01T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="25.3" miny="-78.7" maxx="27.3" maxy="-76.7" time="1995-08-01T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="25.9" miny="-80.0" maxx="27.9" maxy="-78.0" time="1995-08-02T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="26.7" miny="-81.4" maxx="28.7" maxy="-79.4" time="1995-08-02T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="27.2" miny="-82.9" maxx="29.2" maxy="-80.9" time="1995-08-02T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="27.6" miny="-84.4" maxx="29.6" maxy="-82.4" time="1995-08-02T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="27.8" miny="-85.7" maxx="29.8" maxy="-83.7" time="1995-08-03T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="28.3" miny="-86.7" maxx="30.3" maxy="-84.7" time="1995-08-03T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="28.8" miny="-87.6" maxx="30.8" maxy="-85.6" time="1995-08-03T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="29.6" miny="-88.5" maxx="31.6" maxy="-86.5" time="1995-08-03T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="30.4" miny="-89.5" maxx="32.4" maxy="-87.5" time="1995-08-04T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="31.3" miny="-90.1" maxx="33.3" maxy="-88.1" time="1995-08-04T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="32.2" miny="-90.7" maxx="34.2" maxy="-88.7" time="1995-08-04T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="33.1" miny="-91.2" maxx="35.1" maxy="-89.2" time="1995-08-04T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="33.8" miny="-91.2" maxx="35.8" maxy="-89.2" time="1995-08-05T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="34.4" miny="-91.1" maxx="36.4" maxy="-89.1" time="1995-08-05T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="35.3" miny="-90.8" maxx="37.3" maxy="-88.8" time="1995-08-05T12:00Z"/>
<BoundingBox SRS="epsg:4326" minx="36.5" miny="-89.8" maxx="38.5" maxy="-87.8" time="1995-08-05T18:00Z"/>
<BoundingBox SRS="epsg:4326" minx="37.4" miny="-87.8" maxx="39.4" maxy="-85.8" time="1995-08-06T00:00Z"/>
<BoundingBox SRS="epsg:4326" minx="37.7" miny="-85.9" maxx="39.7" maxy="-83.9" time="1995-08-06T06:00Z"/>
<BoundingBox SRS="epsg:4326" minx="37.8" miny="-83.0" maxx="39.8" maxy="-81.0" time="1995-08-06T12:00Z"/>
</LAYER>
--ISRSCOTTWMSVER01010101
Content-Type: video/avi
The encoded movie data.
--ISRSCOTTWMSVER01010101--
```

Just as a proof of concept, mplayer was used on the client side to split the movie into PNG image files, and the existing engine within World Wind displayed the animation frames. The performance increase was surprising, the test animations of 24 frames was loading about 10 frames per minute using all

other implementation models discussed and had a total download size of just over 12MB. Using this model, the first frame was displayed within 10 seconds of the request almost immediately followed by the rest. The download size dropped from just over 12MB to just over 1MB.

It was decided to perform the same test using the Hurricane Katrina Landfall event of 240 frames that took nearly 30 minutes using the other methods and downloaded over 240MB of data. Using our new model the download took less than 2 minutes and has a size of only 5.6MB. The processing time left a little to be desired in that extracting 240 frames took a substantial amount of time on the client; however, it was significantly less than download time required with the other models.

After much testing, it was decided that even with the extra client overhead involved, using a movie encoding technique to transfer the frames of an animation is a viable solution to the animation issues within WMS. The ideal model would be one in which the raw movie file could be played and moved across a geographic path, without decompressing the frames, but even converting the frames to images at the client level showed profound performance improvement over sending the individual frames across the network.

It is important to add that the testing was performed using a 100MB local area network to transmit the data, and these performance numbers would show even greater benefit to the model if tested across a slower link.

9.2 ASE AFD Root Element

An ASE SFD document is defined as a sequence of time-stamped layers. The root element is defined by the WMS layer specification. No modification to the layer definitions is required. The only required modification is to define the mime-type for the movie of the given layer in the Format tag.

9.3 ASE AFD Returned Layer Descriptors

An ASE SFD request will return a multipart/mixed mime-type document. The first part of the document is an XML document based on the Layer class of the WMSCapabilities document type.

Below is an example of a returned XML Document.

```
<?xml version="1.0" encoding="UTF-8"?>
<Layer xmlns="http://www.opengis.net/ase" schemaLocation="http://www.opengis.net/ase/capabilites.xsd">
<!-- First Frame Definition -->
<Layer>
<Dimension name="time" units="ISO8601"/>
<Extent name="time">1995-08-03</Extent>
<Dimension name="frame" units="IS)8601"/>
<Extent name="frame">1</Extent>
<BoundingBox SRS="EPSG:4326" minx="-91" miny="25.24" maxx="-79.36" maxyy="32.20"/>
</Layer>
<!--Frame 2 Definition -->
<Layer>
<Dimension name="time" units="ISO8601"/>
<Extent name="time">1995-08-04</Extent>
<Dimension name="frame" units="IS)8601"/>
<Extent name="frame">2</Extent>
<BoundingBox SRS="EPSG:4326" minx="-90" miny="24.24" maxx="-78.36" maxyy="31.20"/>
</Layer>
<!--Frame3 Definition -->
<Layer>
<!-- Current Frame Definition -->
<Dimension name="time" units="ISO8601"/>
```



```

<Extent name="time">1995-08-05</Extent>
<Dimension name="frame" units="IS)8601"/>
<Extent name="frame">3</Extent>
<BoundingBox SRS="EPSG:4326" minx="-89" miny="23.24" maxx="-77.36" maxyy="30.20"/>
</Layer>
<!--Continue this format for each frame of the animation, even if the frame does not move Geographically, it does still move in time. -->
</Layer>

```

9.4 Example WMS Requests using an ASE AFD (GetAnimation)

9.4.1 General

The GetAnimation operation returns a single frame of an animation. In all respects, it is a map with time-stamped, geographically-rectified image data. Upon receiving a GetAnimation request, a WMS with ASE SFD shall either satisfy the request or issue a service exception.

9.4.2 GetAnimation request overview

The following table describes the GetAnimation Request.

<i>Request Parameter</i>	<i>Mandatory/Optional</i>	<i>Description</i>
VERSION=1.3.0	Mandatory	Request version.
REQUEST=GetFrames	Mandatory	Request name.
LAYER=layer	Mandatory	A single map layer containing an animation. Determined by the presence of an <Layer><Animation> element in the service metadata.
CRS=namespace:identifier	Optional (Default CRS of the server will be returned as part of the XML document.)	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	Optional (Default Bounding Box of the frame will be returned as part of the XML document.)	Bounding box corners (lower left, upper right) in CRS units
FORMAT=output_format	Mandatory	Output format of the animation.
TRANSPARENT=TRUE/FALSE	Optional	Background transparency of the animation (default=FALSE)
BGCOLOR=color_value	Optional	Hexadecimal red-green-blue color value for the background color (default=0xFFFFFF)
EXCEPTIONS=exception_format	Optional	The format in which exceptions are to be reported by the WMS (default=XML)
TIME=time	Optional	Time values of the requested frames. This option may not be support by the server implementation in which case the server will ignore the time option and return all frames of the animation sequence.

<i>Request Parameter</i>	<i>Mandatory/Optional</i>	<i>Description</i>
ELEVATION=elevation	Optional	Elevation of the layer/frame requested.
Other sample dimension(s)	Optional	Value of other dimensions as appropriate.

9.4.3 Request Parameters

9.4.3.1 VERSION

The mandatory VERSION parameter is defined in 6.9.1 of the OpenGIS Web Map Service Implementation Specification version 1.3 OGC Document number OGC 04-024. The value “1.3.0” shall be used for GetFrames requests that comply with this International Standard.

9.4.3.2 REQUEST

The mandatory REQUEST parameter is defined in 6.9.2 of the OpenGIS Web Map Service Implementation Specification version 1.3 OGC Document number OGC 04-024. To invoke the GetAnimation operation, the value “GetAnimation” shall be used.

9.4.3.3 LAYER

The mandatory LAYER parameter is the map layer to be returned by the GetAnimation request. The value of the LAYER parameter is a valid layer name. Allowed layer names are the character data content of any <Layer><Name> element in the service metadata that has an associated <Layer><Animation> element.

9.4.3.4 CRS

The CRS request parameter states which Layer CRS (Section 6.7.3 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024) applies to the BBOX request parameter. The value of the CRS parameter in a request to a particular server shall be one to the values defined in that server's service metadata in a <CRS> element defined or inherited by the requested layer.

For more information on the CRS parameter, refer to section 7.3.3.5 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

9.4.3.5 BBOX

The optional BBOX parameter allows a Client to request a particular Bounding Box (or region of a map layer.) The details of the BBOX parameter are outlined in section 7.3.3.6 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024.

9.4.3.6 FORMAT

The mandatory FORMAT parameter states the desired format of the frame/map. Support values for a GetAnimation request on a WMS server are listed in one or more <Request> <GetAnimation> <Format> elements of its service metadata. The entire MIME type string in <Format> is used as the

value of the FORMAT parameter. There is no default format; however, the FORMAT parameter only affects the embedded image format. The return type for a frames is always *multipart/mixed* as the frames contain default bounding box and time parameters in a preceding xml document for each frame element. In an HTTP environment, the MIME type shall be set on the returned object using the Content-type entity header, and it shall be *multipart/mixed*. The requested FORMAT MIME type will be embedded in the multipart/mixed data as a subtype using the Content-type entity header. If the request specifies a format not supported by the server, the server shall issue a service exception (code=InvalidFormat).

9.4.3.7 TIME

The optional TIME parameter is defined as a time value to be expressed as defined in Section 6.7.5 and Annexes C and D of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024. For the GetAnimation request, the time parameter is used by the server to extract frames from the full animation. It need only be optionally implemented on the server. If the option is implemented the server will return a rendered animation containing only the requested frame, otherwise the server shall return the full animation along with the frame parameters as defined in section 10.3.

9.4.3.8 Other Optional Parameters

The additional optional parameters are clearly defined in the subsections of section 7.3 of the *OpenGIS Web Map Service Implementation Specification version 1.3* OGC Document number OGC 04-024. The options are used in exactly the same manner as they are for GetMap requests.

Annex A Compliance Testing

(Normative)

A.1 GetFrames Response

Verify that a valid map image and metadata is returned for several test case urls.

Returned data should be multi-part MIME type of the following format.

Content-Type: multipart/mixed; boundary=UNIQUE_SEPERATION_IDENTIFIER_STRING

--UNIQUE_SEPERATION_IDENTIFIER_STRING

Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="UTF-8"?>

<Layer xmlns="http://www.opengis.net/ase" schemaLocation="http://www.opengis.net/ase/capabilites.xsd">

<!-- First Frame Definition -->

<Layer>

<Dimension name="time" units="ISO8601"/>

<Extent name="time">1995-08-03</Extent>

<BoundingBox SRS="EPSG:4326" minx="-91" miny="25.24" maxx="-79.36" maxyy="32.20"/>

</Layer>

--UNIQUE_SEPERATION_IDENTIFIER_STRING

Content-Type: image/png

The MIME encoded image data

--UNIQUE_SEPERATION_IDENTIFIER_STRING

Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="UTF-8"?>

<Layer xmlns="http://www.opengis.net/ase" schemaLocation="http://www.opengis.net/ase/capabilites.xsd">

<!-- First Frame Definition -->

<Layer>

<Dimension name="time" units="ISO8601"/>

<Extent name="time">1995-08-03</Extent>

<BoundingBox SRS="EPSG:4326" minx="-91" miny="25.24" maxx="-79.36" maxyy="32.20"/>

</Layer>

--UNIQUE_SEPERATION_IDENTIFIER_STRING

Content-Type: image/png

The MIME encoded image data

The above sequence will be repeated for each frame of the animation that was requested.

--UNIQUE_SEPERATION_IDENTIFIER_STRING--

A.2 GetAnimation Response

Verify that a valid movie file and metadata is returned for several test case urls.

Returned data should be multi-part MIME type of the following format.

Content-Type: multipart/mixed; boundary=UNIQUE_SEPERATION_IDENTIFIER_STRING

--UNIQUE_SEPERATION_IDENTIFIER_STRING

Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="UTF-8"?>

<Layer xmlns="http://www.opengis.net/ase" schemaLocation="http://www.opengis.net/ase/capabilites.xsd">

<!-- First Frame Definition -->

<Layer>

<Dimension name="time" units="ISO8601"/>

<Extent name="time">1995-08-03</Extent>

<Dimension name="frame" units="IS)8601"/>

<Extent name="frame">1</Extent>

<BoundingBox SRS="EPSG:4326" minx="-91" miny="25.24" maxx="-79.36" maxyy="32.20"/>

</Layer>

<!-- Frame 2 Definition -->

<Layer>

<Dimension name="time" units="ISO8601"/>

<Extent name="time">1995-08-04</Extent>

<Dimension name="frame" units="IS)8601"/>

<Extent name="frame">2</Extent>

<BoundingBox SRS="EPSG:4326" minx="-90" miny="24.24" maxx="-78.36" maxyy="31.20"/>

```
</Layer>
<!--Frame3 Definition -->
<Layer>
<!-- Current Frame Definition -->
<Dimension name="time" units="ISO8601"/>
<Extent name="time">1995-08-05</Extent>
<Dimension name="frame" units="IS)8601"/>
<Extent name="frame">3</Extent>
<BoundingBox SRS="EPSG:4326" minx="-89" miny="23.24" maxx="-77.36" maxyy="30.20"/>
</Layer>
<!--Continue this format for each frame of the animation, even if the frame does not move Geographically, it does still move in time. -
->
</Layer>
--UNIQUE_SEPERATION_IDENTIFIER_STRING
Content-Type: video/avi
The MIME encoded movie data
--UNIQUE_SEPERATION_IDENTIFIER_STRING--
```

Annex B: Animation Service Extension Schema

(Normative)

This annex contains the Animation Service Extension Schema corresponding to this version of the specification. Comments in the schema are informative; in case of conflict with the main body of this specification, the main body takes precedence. The schema definitions here (without comments) is normative; in case of conflict with schema definitions in the main body of this specification, the schema takes precedence.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!--W3C Schema generated by XMLSpy v2006 sp2 U (http://www.altova.com)-->
<xsd:schema xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wms="http://www.opengis.net/wms"
xmlns:ase="http://www.opengis.net/ase" targetNamespace="http://www.opengis.net/ase" elementFormDefault="qualified">
  <xsd:import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://schemas.opengis.net/gml/2.1.1/xlinks.xsd"/>
  <xsd:import namespace="http://www.opengis.net/ogc"
schemaLocation="http://schemas.opengis.net/filter/1.0.0/filter.xsd"/>
  <xsd:import namespace="http://www.opengis.net/wms"
schemaLocation="http://schemas.opengis.net/wms/1.3.0/capabilities_1_3_0.xsd"/>
  <!--*****_-->
  <xsd:annotation>
    <xsd:documentation>
      Animation Service Extension version 0.1.0 (2005-12-07)
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="GetAnimation">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="wms:Format"/>
        <xsd:element ref="wms:Format"/>
        <xsd:element ref="wms:DCPType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="GetFrames">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="wms:Format"/>
        <xsd:element ref="wms:Format"/>
        <xsd:element ref="wms:DCPType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Request">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ase:GetFrames"/>
        <xsd:element ref="ase:GetAnimation"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```