

WFS Design Document

June 18, 2004



Submitted To: Program Manager
GeoConnections
Victoria, BC, Canada

Submitted By: David Zwiers
Refractions Research Inc.
Suite 400 - 1207 Douglas Street
Victoria, BC V8W 2E7
E-mail: dzwiers@refractions.net
Phone: (250) 383-3022
Fax: (250) 383-2140

TABLE OF CONTENTS

INTRODUCTION.....	3
2 WFS DATASTORE DESIGN.....	4
2.1 MODULE OVERVIEW	4
2.2 GMLDATASTORE.....	5
2.3 WFSDATASTORE.....	6
3 GML.....	8
3.1 GML PARSER	8
3.2 GML WRITER.....	10

TABLE OF FIGURES

Figure 1 - GMLDataStore Class Diagram.....	5
Figure 2 - WFSDataStore Class Diagram.....	7
Figure 3 - Parser Class Diagram	9
Figure 4 - Writer Class Diagram.....	10

1 INTRODUCTION

This document illustrates our design for the Web Feature Server (WFS) data access component of the uDig project. This portion of the uDig project makes extensive use of the DataStore interfaces developed within the GeoTools library.

This document includes three main portions

- Introduction to the Geotools DataStore API
- GMLDatastore / WFSDataStore Design
- GML Parser Design

All of these components will be implemented with the assistance of the Geotools Open Source Community. Upon their completion, the components outlined in this document will reside within the Geotools codebase.

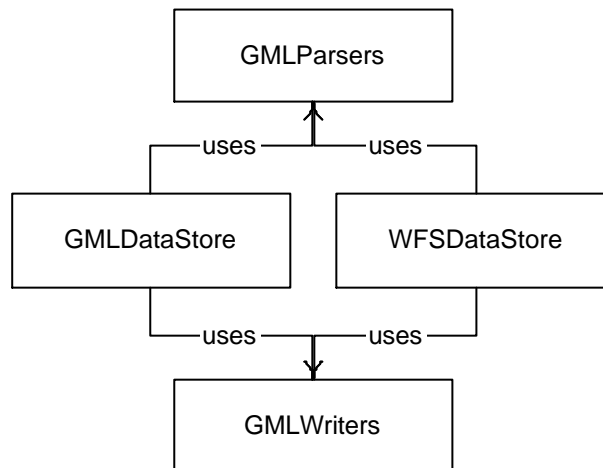
The GMLDataStore and the WFSDataStore implementations will both adhere to the OGC GML 2.1 specification. The WFSDataStore will also adhere to the OGC WFS Specification.

2 WFS DATASTORE DESIGN

For the upcoming uDig project we intend to create a WFS DataStore following the GeoTools DataStore API. This document provides an overview of the design for the proposed DataStore, as well as documenting the effects this will have on other GeoTools modules.

2.1 Module Overview

Currently GML is parsed using the GMLDataSource. It is my understanding that this is to be phased out. As such I intend to create a new module with a set of GML parsers, GML writers, a GMLDataStore and a WFSDataStore. An overview of the module interactions can be seen in the following diagram:



As can be seen in the preceding diagram, the GMLDataStore and WFSDataStore both use the GML parsers and writers. We can also see that there is not a functional dependence between the parsers and writers, nor between the GMLDataStore and the WFSDataStore.

2.2 GMLDataStore

The GMLDataStore is intended to provide access to GML parser/writer in a convenient way. For the most part, the GMLDataStore will be a wrapper for the GML parsers and writers. The GMLDataStore is intended to be OGC GML 2.1 compliant.

GMLDataStore functional requirements:

- GML 2.1 Compliant
- Reads/Writes from a URL
- Parses/Writes results made available via GMLFeatureReader / GMLFeatureWriter

2.2.1 GMLDataStore Class Diagram

We envision the GMLDataStore module class structure as depicted in the following Figure.

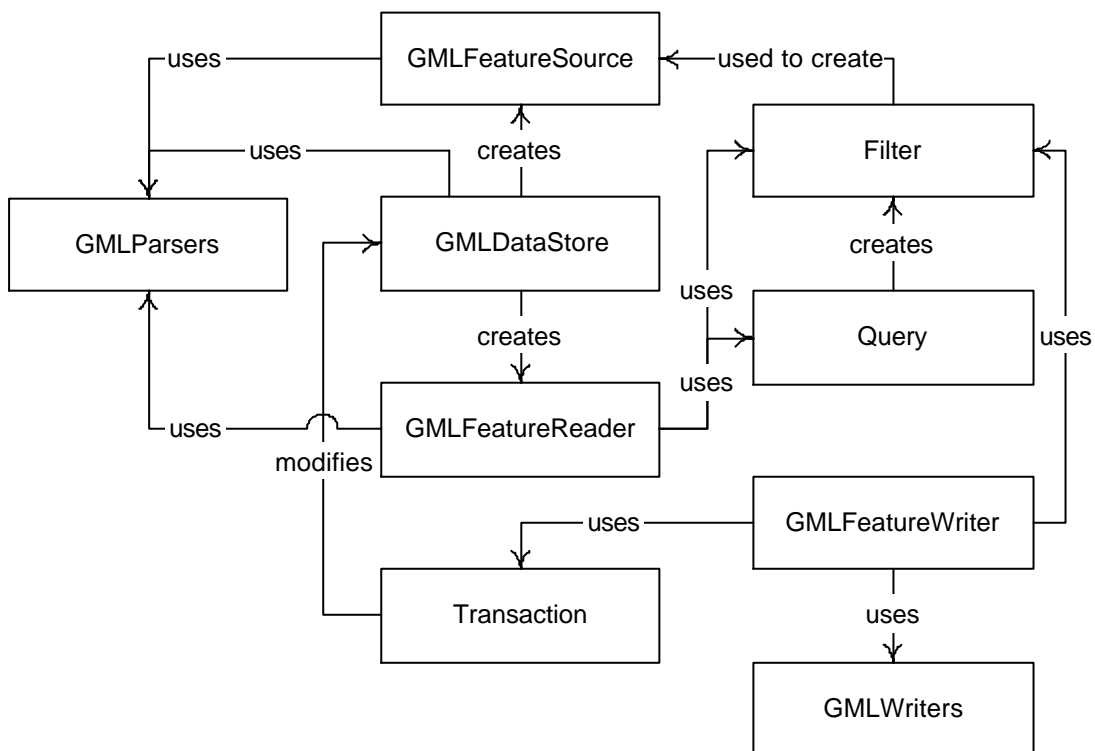


Figure 1 - GMLDataStore Class Diagram

2.3 WFSDataStore

The WFSDataStore is a new module intended to communicate with OGC compliant WFS servers. This module will make extensive use of the GML parsers, and will implement the DataSource API.

WFSDataStore functional requirements:

- WFS 1.0.0, GML 2.1 Compliant
- Reads and Writes to a WFS
- Parses and Writes WFS Requests
 - GetCapabilities
 - DescribeFeatureType
 - GetFeature
 - LockFeature
 - Transaction

WFSDataStore non-functional requirements:

- Scalable
- Streamable
- Self-initializing

2.3.1 WFSDataStore Class Diagram

We envision the WFSDataStore module class structure as depicted in the following Figure.

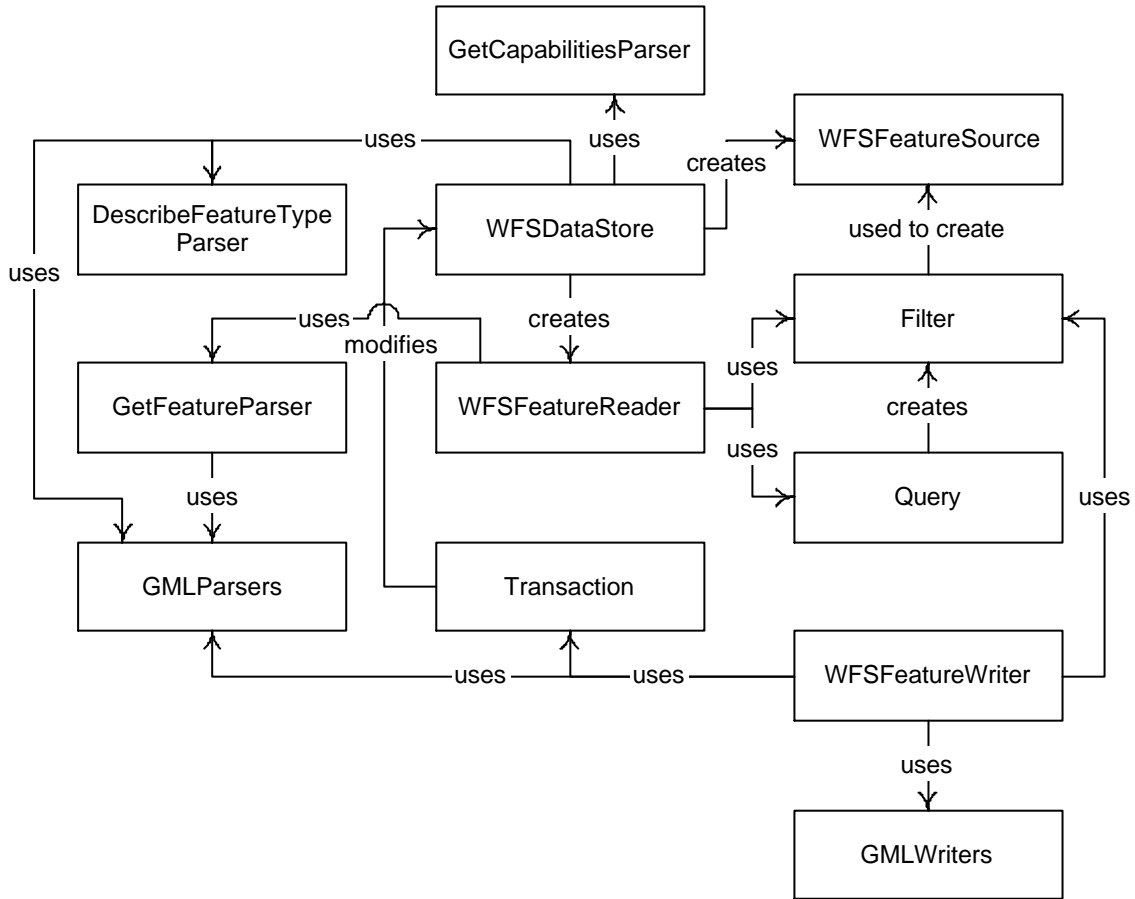


Figure 2 - WFSDataStore Class Diagram

3 GML

3.1 GML Parser

For the use of GML DataStore and WFS DataStore our parser is required to handle the following:

- Feature
- Geometry
- Filter

To further handle the needs of a WFS DataStore our parser will be extended to handle:

- FeatureType

Our parser can be extended with additional GML 2.1 constructs as required.

Other non-functional development goals for this module include:

- Scalable
- Streamable

3.1.1 Parser Class Diagram

We envision the GML Parser module class structure as depicted in the following Figure..

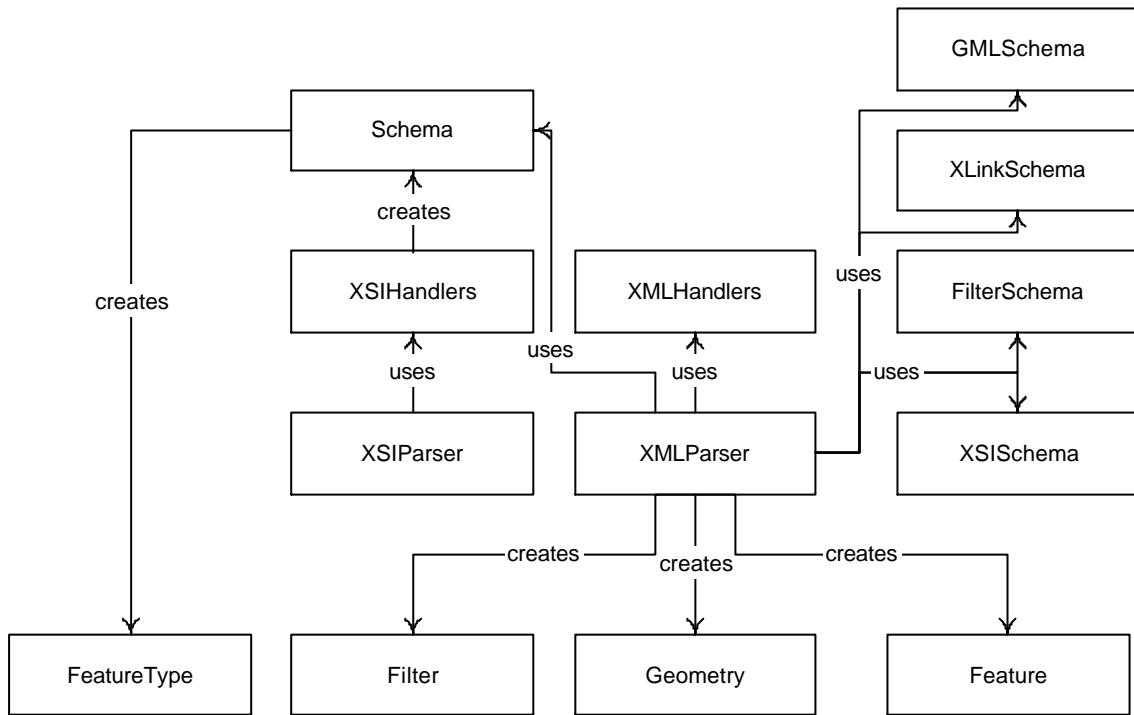


Figure 3 - Parser Class Diagram

3.2 GML Writer

Although a complete GML Writer is not required, our parser should as a minimum print the following GML 2.1 constructs:

- Feature
- Geometry
- Filter

The GeoServer project will require this portion of the GeoTools library provide:

- FeatureType

Other non-functional development goals for this module include:

- Scalable
- Streamable

3.2.1 Writer Class Diagram

We envision the GML Writer module class structure as depicted in the following Figure.

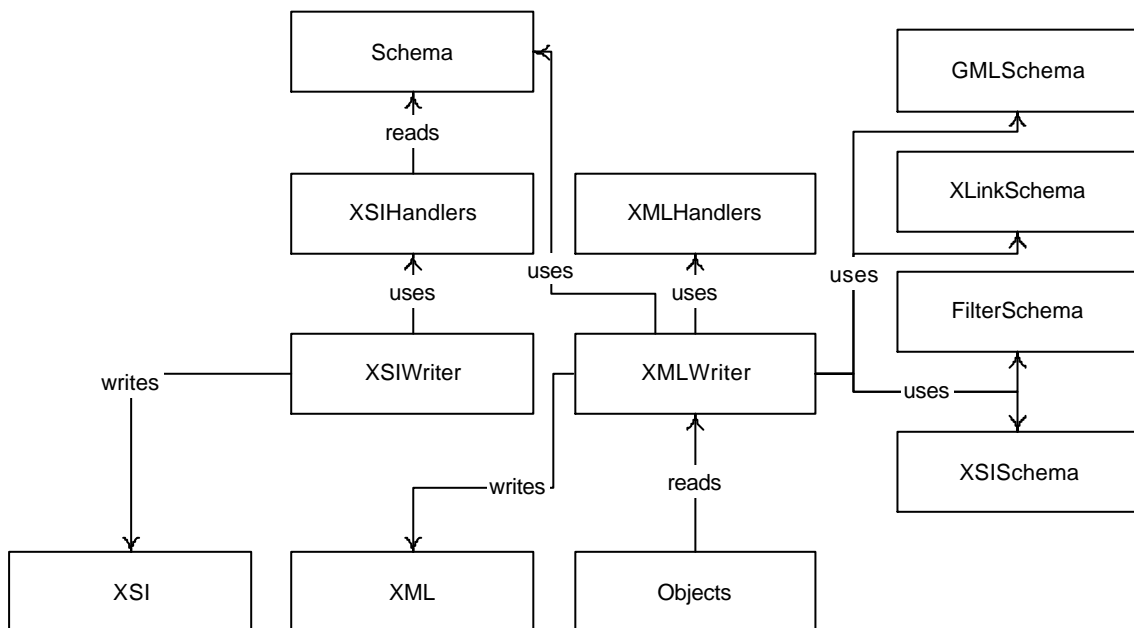


Figure 4 - Writer Class Diagram