May 2011
Oracle Spatial User Conference
Oracle Spatial User Conference

May 19, 2011
Ronald Reagan Building and International Trade Center
Washington, DC  USA
Albert Godfrind
Spatial Solutions Architect
Oracle
Oracle Spatial Web Services
WMS, WFS and more
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
What We Will Cover

✓ Web Map Service (WMS)
  ✓ Licensing, configuration, example requests
✓ Web Feature Service (WFS)
  ✓ Licensing, configuration, example requests
✓ Using OGC Web Services in Mapviewer
✓ Other services
Open Geospatial Consortium Standards

- Web Map Service (WMS)
  - Request/Provide maps
  - Request/Provide information about content of a map

- Web Feature Service (WFS)
  - Request/Provide data
  - Access/search/update/delete geo-spatial feature instances
  - Based on spatial/non-spatial search criteria
  - Using a standard interface over the web
Why is this important?

- Sharing spatial information between users
- Protects applications from infrastructure details
- A fundamental building block for enabling cooperation between multiple public and private organisations
- Example: a fire brigade needs information from multiple sources
  - Property boundaries from city administration
  - Gas pipes from utilities
  - Traffic from live sensors
- A requirement for all public administrations in the EU
- INSPIRE directive
WMS Web Mapping Service
Web Mapping Service

- **Provided by Mapviewer**
  - Need to license Oracle Fusion Middleware
  - Can also license the “TopLink and Application Development Framework” for deployment in other application servers

- **OGC Conformance**
  - Versions 1.1.1 and 1.3.0

- **Processes WMS requests:**
  - GetCapabilities
  - GetMap
  - GetFeatureInfo
### Configuring WMS

- In MapViewer’s configuration file
- Set descriptive information for the service:
  - Title and description (abstract) of the service
  - Keywords associated with the service
  - Will be returned in the capabilities document

```xml
<wms_config>
    <title>
        WMS 1.1 interface for MapViewer
    </title>
    <abstract>
        This WMS service is provided through Oracle MapViewer.
    </abstract>
    <keyword_list>
        <keyword>roadrunner</keyword>
        <keyword>ambush</keyword>
    </keyword_list>
</wms_config>
```
Configuring WMS

• Set explicit service URL
  • Returned in the capabilities document
  • Host, port, protocol

• Set the default datasource
  • Used when the GetMap request specifies no data source
  • If no default set, then GetMap will use a datasource called “WMS”

```xml
<wms_config
  host="www.oracle.com"
  port="80"
  protocol="http"
  default_datasource="mvdemo" >
</wms_config>
```
Configuring WMS

- Set coordinate system ID (SRID) mappings
- Specify the name of a mapping file

```xml
<wms_config>
  <sdo_epsg_mapfile>
    ../conf/epsg_srids.properties
  </sdo_epsg_mapfile>
</wms_config>
```

- For example, the following associates Oracle SRIDs 82208 and 82771 with their EPSG equivalents

```plaintext
82208=32601
82271=32602
```

- Note that Oracle SRID 8307 (Longitude Latitude WGS84) is automatically associated with EPSG SRID 4326.
GetCapabilities request

http://127.0.0.1:7001/mapviewer/wms
?VERSION=1.1.0&REQUEST=GetCapabilities&SERVICE=WMS

<WMT_MS_Capabilities version="1.1.1">
  <Service>
    <Name>OGC:WMS</Name>
    <Title>
      WMS 1.1 interface for MapViewer
    </Title>
    <OnlineResource xlink:type="simple"
      xlink:href="http://127.0.0.1:7001/mapviewer/wms"/>
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
  </Service>
</WMT_MS_Capabilities>

- Lists all MapViewer “themes” as layers
- Returns coordinate systems as EPSG codes
GetMap request

- Specify coordinate systems using EPSG codes
  - Mapping table in Mapviewer configuration file
- Specify the list of layers to include
  - They correspond to Mapviewer themes
  - List them in the order of rendering

http://127.0.0.1:7001/mapviewer/wms
?VERSION=1.1.0
&REQUEST=GetMap
&FORMAT=image/png
&SRS=EPSG:4326
&BBOX=-80,38,-78,40
&WIDE=480
&HEIGHT=400
GetMap request

- Coordinate system (SRS)
  - Used to specify the coordinates of the bounding box
  - Also used to specify the coordinate system of the resulting map
  - Layer results will be automatically transformed to this coordinate system!

http://127.0.0.1:7001/mapviewer/wms
?VERSION=1.1.0
&REQUEST=GetMap
&FORMAT=image/png
&SRS=EPSG:3785
&BBOX=-8905559.3,4579425.81,-8682920.3,4865942.28
&WIDTH=480
&HEIGHT=400
GetMap request
MapViewer-specific parameters

• Specify a base map
  • Corresponds to a Mapviewer base map
  • Used as a backdrop for explicit layers (specified by LAYERS=)

• Specify an explicit datasource
  • If none specified: use the default datasource

```
http://127.0.0.1:7001/mapviewer/wms
?VERSION=1.1.0
&REQUEST=GetMap
&FORMAT=image/jpeg
&SRS=SDO:8307
&BBOX=-126,33,-114,45
&WIDTH=480
&HEIGHT=400
&BASEMAP=US_BASE_MAP
&DATASOURCE=spatial
&LAYERS=US_PARKS
```
GetFeatureInfo request

- Same parameters as GetMap
- Specify layer (= theme) to query
- Results contain the “info” columns specified for that theme
- Can also specify Mapviewer-specific query parameters
  - Query_type=at_point, nn, within_distance

```
http://127.0.0.1:7001/mapviewer/wms
?VERSION=1.1.0
&REQUEST=GetFeatureInfo
&query_Layers=US_STATES
&SRS=SDO:8307
&BBOX=-126,33,-114,45
&WIDTH=480
&HEIGHT=400
&X=200
&y=200
```

```
<GetFeatureInfo_Result>
  <ROWSET name="US_STATES">
    <ROW num="1">
      <STATE>California</STATE>
      <STATE_ABRV>CA</STATE_ABRV>
      <TOTPOP>29760016</TOTPOP>
      <POPPSQMI>190.8347</POPPSQMI>
    </ROW>
  </ROWSET>
</GetFeatureInfo_Result>
```
WFS Web Feature Service
Web Feature Service

- **Provided with Oracle Spatial**
  - Need to license the Oracle Spatial option
- Can deploy in any application server
  - 11gR1: only certified with Oracle Application Server
  - 11gR2: certified with Oracle Weblogic Server
- **OGC Conformance**
  - Versions 1.0
- Processes all WFS requests:
  - Basic
  - Transactional (WFS-T)
- SOAP and XML interfaces
Web Feature Service Overview

- OGC standard (and ISO 19142)
  - current version 1.1
- A WFS is a Web interface that allows one to publish and deploy geographic feature data locally or across the Internet.
- Two types of services:
  - “standard” WFS
  - “transactional” WFS (WFS-T)
- WFS-T operations include the ability to insert, delete, update, get and query features on spatial and non-spatial constraints
Web Feature Service Architecture

Client

WFS Client

WFS Requests/responses

HTTP

Application Server

WFS Server API

Feature cache

JDBC

Database

Spatial Tables

Feature Mapping

Admin API (PL/SQL)
Installing the WFS

✓ Deploy the SDOWS web application
✓ Configure database accounts
  • For administration and data access
✓ Configure data sources
  • Access to those accounts from the web applications
✓ Set web service parameters
  • Logging, timeouts, etc
  • File WEB-INF/conf/wsconfig.xml
Logging and debugging

- Set logging level to “finest” while setting up and testing your web services

```xml
<logging log_level="finest" log_thread_name="true" log_time="true">
  <log_output name="System.err" />
  <log_output name="log/ws.log" />
</logging>
```

- All logging and debugging information written to file `WEB-INF/log/ws.log`
- Highly detailed information: all requests and responses, all SQL statements, exceptions, etc
- Essential for debugging problems
Configuring the WFS

- All administration steps performed in SQL
- Packages SDO_WFS_PROCESS and SDO_WFS_LOCK

1. Set capabilities info
2. Enable the schema
3. Publish table
4. Register table for updates
5. Notify WFS server
6. Grant access rights
7. Set configuration params
1. Set server capabilities info

- Sets the base content of the information returned by the `GetCapabilities` request:
  - Name and description of the server
  - List of supported requests
  - List of supported OGC filter operations
- Provide it as an XML document
- Really a template that will be filled with actual data by the `GetCapabilities` request
  - URLs will be replaced with the actual server URLs
  - Published feature types will be added
Set server capabilities info

- Load the capabilities info from an XML file
  - Define a directory that points to the XML file:

```sql
create or replace directory WFS_XML
as '<directory that contains the template>';```

- Load the file using the "BFILE" mechanism:

```sql
begin
  SDO_WFS_PROCESS.insertCapabilitiesInfo(
    xmltype(
      bfilename('WFS_XML', 'WFS_capabilitiesTemplate.xml'),
      nls_charset_id('AL32UTF8')
    )
  );
end;
/
commit;
```
2. Enable Database Schemas

- Schemas that contain tables to be published must be enabled.
- This allows data in those schemas to be published via WFS

```sql
SQL> exec SDO_WFS_PROCESS.GrantMDAccessToUser('SCOTT');
```

- To revoke access:

```sql
SQL> exec SDO_WFS_PROCESS.RevokeMDAccessFromUser('SCOTT');
```
3. Publish Tables as Feature Types

• Perform this step for all tables or views that you want to publish to WFS

• Use `SDO_WFS_PROCESS.publishFeatureType()`
  • Inserts the definition of the table or view in the metadata repository used by WFS
  • Grants all rights on the table to `SPATIAL_WFS_ADMIN`
Publish a Database Table

```
declare
  featureDescriptorXML CLOB :=
    '<?xml version="1.0" ?>
    <FeatureType xmlns:scottns="http://www.myserver.com/scott"
      xmlns="http://www.opengis.net/wfs">
      <Name> scottns:US-Cities</Name>
      <Title>U.S Cities</Title>
      <SRS>SDO:8307</SRS>
    </FeatureType>';
begin
  SDO_WFS_PROCESS.publishFeatureType(
    dataSrc => 'SCOTT.US_CITIES',
    ftNsUrl => 'http://www.myserver.com/scott',
    ftName => 'UsCities',
    ftNsAlias => 'scottns',
    featureDesc => xmltype(featureDescriptorXML),
    schemaLocation => null,
    pkeyCol => 'ID',
    columnInfo => MDSYS.StringList('PointMemberType'),
    pSpatialCol => 'LOCATION',
    featureMemberNs => null,
    featureMemberName => null,
    srsNs => null,
    srsNsAlias => null
  );
end;
/```
Publish Multiple Tables

- Use PL/SQL
- Identify the tables to publish
  - For example, all tables in a schema
- For each table
  - Get the name of the primary key column
  - Dynamically generate the feature descriptor
  - Publish it
begin
  for t in (select g.owner, g.table_name, g.column_name, g.srid
          from all_sdo_geom_metadata g, all_tab_cols c
          where g.owner = 'SCOTT' and c.owner = g.owner
          and c.table_name = g.table_name and c.column_name = g.column_name
          and c.data_type = 'SDO_GEOMETRY')
  loop
    -- Construct feature descriptor
    featureDescriptorXML :=
      '<?xml version="1.0" ?>' ||
      '  <FeatureType xmlns:myns="http://www.myserver.com/myns" ||
       xmlns="http://www.opengis.net/wfs">' ||
      '    <Name> ' || namespace_alias || ':' || t.table_name || '</Name>' ||
      '    <Title>Database Table ' || t.table_name || '</Title>' ||
      '    <SRS>SDO:' || t.srid || '</SRS>' ||
      '</FeatureType>';
-- Publish the type
SDO_WFS_PROCESS.publishFeatureType(
    dataSrc => t.owner || '.' || t.table_name,
    ftNsUrl => p_namespace_url,
    ftName => t.table_name,
    ftNsAlias => p_namespace_alias,
    featureDesc => xmltype(featureDescriptorXML),
    schemaLocation => null,
    pkeyCol => pkey,
    columnInfo => MDSYS.StringList('GeometryCollectionType'),
    pSpatialCol => t.column_name,
    featureMemberNs => null,
    featureMemberName => null,
    srsNs => null,
    srsNsAlias => null
);

end loop;
end;
List Published Feature Types

• List all feature types published in a database schema:

```sql
select featureTypeName, dataPointer, namespacePrefix, namespaceURL
from mdsys.WFS_FeatureType$
where dataPointer like 'SCOTT.%'
order by featureTypeId;
```

• List feature types published by the current schema:

```sql
select featureTypeName, dataPointer, namespacePrefix, namespaceURL
from mdsys.WFS_FeatureType$
where dataPointer like user || '.%'
order by featureTypeId;
```
Unpublish a Feature Type

• Use the **SDO_WFS_PROCESS.dropFeatureType()** procedure to unpublish one specific type.
• Must “unpublish” first before re-publishing
  • For example, if your table structure changed

```sql
begin
    SDO_WFS_PROCESS.dropFeatureType(
        'http://www.myserver.com/scott', 'UsCities' );
end;
/
```
Unpublish All Feature Types in a Schema

begin
    for f in (select featureTypeId, featureTypeName, dataPointer, namespaceURL
                from mdsys.WFS_FeatureType$
                where dataPointer like 'SCOTT.%'
            )
    loop
        SDO_WFS_PROCESS.dropFeatureType(f.namespaceURL, f.featureTypeName);
        dbms_output.put_line ('Table ' || f.dataPointer || ' unpublished');
    end loop;
end;
/
commit;
4. Register Tables for Updating

- Database tables updated in WFS-T transactions must be registered
- Registering a feature table enables the table for WFS transaction locking
- Run this as the user that owns the tables

```
SQL> exec SDO_WFS_LOCK.registerFeatureTable('SCOTT', 'US_CITIES');
```

- To un-register the table:

```
SQL> exec SDO_WFS_LOCK.unRegisterFeatureTable('SCOTT', 'US_CITIES');
```
5. Notify the WFS server

- Changes are not automatically visible by the running WFS
- Use **SDO_WFS_PROCESS.insertFtMDUpdated()** to notify the WFS

```sql
exec SDO_WFS_PROCESS.InsertFtMDUpdated('http://www.myserver.com/scott','UsCities', sysdate);
```

- The WFS checks for notification at regular intervals
- Based on **wfs_cache_sync_interval** parameter
  - Default is 10 seconds

```xml
<wfs_cache_sync_interval>10000</wfs_cache_sync_interval>
```
## Using the WFS

### BASIC - Data Publishing

- **GetCapabilities**
  - Gets the metadata about the types / operations that the feature server supports
- **DescribeFeature**
  - Gets the structural information about a feature type
- **GetFeature**
  - Gets a set of features

### WFS-T – Data Editing

- **GetFeatureWithLock**
  - Get a set of features, and lock them for a certain period of time.
- **LockFeature**
  - Lock a set of feature instances
- **Transaction**
  - Insert, update, delete feature instances
WFS Operations – Basic Data Publishing

- **GetCapabilities**
  - Gets the metadata about the types / operations that the feature server supports
- **DescribeFeature**
  - Gets the structural information about a feature type
- **GetFeature**
  - Gets a set of features
GetCapabilities request

Two interfaces are possible

• Annotated URL ("key-value pairs" - KVP)

http://host:port/SpatialWS-SpatialWS-context-root/xmlwfsservlet
?request=GetCapabilities
&service=WFS
&version=1.0.0

• XML request

<GetCapabilities
    service="WFS" version="1.0.0" xmlns="http://www.opengis.net/wfs" />


GetCapabilities response

- Response formed using the template you provided
- Service URLs filled in
- Lists all published feature types

```
<WFS_Capabilities xmlns="http://www.opengis.net/wfs" version="1.0.0"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:myns="http://www.myserver.com/myns">
    <Service>
        <Name> Oracle WFS </Name>
        <Title> Oracle Web Feature Service </Title>
        <Abstract> Web Feature Service maintained by Oracle </Abstract>
        <OnlineResource>
        </OnlineResource>
    </Service>

    <FeatureType xmlns:myns="http://www.myserver.com/scott">
        <Name>scottns:UsStates </Name> <Title> US States </Title> <SRS> SDO:8307</SRS>
    </FeatureType>

    ...</WFS_Capabilities>
```
DescribeFeatureType request

```xml
<wfs:DescribeFeatureType
    service="WFS"
    version="1.0.0"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:myns="http://www.myserver.com/myns"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/WFS-basic.xsd">
  <wfs:TypeName>scottns:UsCities</wfs:TypeName>
</wfs:DescribeFeatureType>
```

Feature type to describe
DescribeFeatureType response

```xml
  <xsd:element name="UsCities" type="scottns: UsCitiesType" substitutionGroup="gml:_Feature"/>
  <xsd:complexType name="UsCities Type">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="ID" type="xsd:double"/>
          <xsd:element name="CITY" nillable="true">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="42"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="STATE_ABRV" nillable="true">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="2"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="POP90" type="xsd:double" nillable="true"/>
          <xsd:element name="RANK90" type="xsd:double" nillable="true"/>
          <xsd:element name="LOCATION" type="gml:GeometryCollectionType" nillable="true"/>
        </xsd:sequence>
        <xsd:attribute name="fid" type="xsd:double"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Schema (xsd) of the feature type
GetFeature request

- Use the OGC Filter syntax
- Flexible query expression (spatial, non spatial, boolean, ...)

```xml
<wfs:GetFeature
    service="WFS"
    version="1.0.0"
    handle="Example Query"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:scottns="http://www.myserver.com/scott">
    <wfs:Query typeName="scottns:UsCities">
        <ogc:PropertyName>scottns:STATE_ABRV</ogc:PropertyName>
        <ogc:PropertyName>scottns:CITY</ogc:PropertyName>
        <ogc:Filter>
            <ogc:Within>
                <ogc:Property Name="scottns:LOCATION">
                    <gml:Box srsName="SDO:8307">
                        <gml:coordinates>105.11, 38.86 -104.72, 39.76</gml:coordinates>
                    </gml:Box>
                </ogc:Within>
            </ogc:Filter>
        </wfs:Query>
    </wfs:GetFeature>
```
GetFeature Response

</wfs:FeatureCollection>
WFS Operations – Transactional Data Editing

- **GetFeatureWithLock**
  - Get a set of features, and lock them for a certain period of time.
- **LockFeature**
  - Lock a set of feature instances
- **Transaction**
  - Insert new feature instances
  - Update existing feature instances based on filter criteria
  - Delete existing feature instances based on filter criteria
  - Can issue multiple update/insert/delete operations in a single transaction request

- Tables to be edited must have been previously registered using `SDO_WFS_LOCK.registerFeatureTable()`
The WFS Transaction and Locking Model

- You need to hold a « token » in order to update selected features
- Must obtain this token before updating features
  - Grants exclusive access to a specific set of features
- Present the token when updating those features
- Tokens have a limited life time
  - Use them before they expire.

(You can also perform updates without tokens!)
The WFS Transaction and Locking Model

*Implementation: « Locks »*

- Lock features while reading (*GetFeatureWithLock*), or lock them explicitly without reading (*LockFeature*)
  - Present the returned lock id for subsequent updates of locked features
- Locks are persistent and last for a specified duration
  - Duration expressed in minutes. The default is specified in the Web Services Configuration – **4 minutes** if nothing specified
- Locks are exclusive: only one requestor can hold a lock on a feature
- Reads are still possible without any conflict (*GetFeature*)
- No “unlock”, “end-of-transaction”, “commit” or “rollback” to release locks before they expire
  - Must wait until locks expire!
Performing Updates and Deletes

“Direct” updates

- Use the **Transaction Update** or **Transaction Delete** operations *without* specifying any **LockId**
  - Will fail if anyone has locked one or more features to update
  - Features are not locked after the update completes

“Transactional” updates

- First lock the features with a **GetFeatureWithLock** or **LockFeature** operation
  - Then use the **Transaction Update** or **Transaction Delete** operations specifying the **LockId** you obtained
- Will fail if features not locked by that lock id
  - (You forgot to lock the features or the lock expired)
  - Features remain locked after the update completes
Locking Configuration Parameters

• Default lock duration (in minutes)

<wfs_lock_expiry>4</wfs_lock_expiry>

• Lock timeout (in seconds)

<wfs_query_timeout>10</wfs_query_timeout>
<wfs:GetFeatureWithLock
    service="WFS"
    version="1.0.0"
    expiry="5"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:scottns="http://www.myserver.com/scott">
    <wfs:Query typeName="scottns:UsCities">
        <ogc:PropertyName>scottns:STATE_ABRV</ogc:PropertyName>
        <ogc:PropertyName>scottns:CITY</ogc:PropertyName>
        <ogc:Filter>
            <ogc:Within>
                <ogc:PropertyName>scottns:LOCATION</ogc:PropertyName>
                <gml:Box srsName="SDO:8307">
                    <gml:coordinates>-105.11, 38.86 -104.72, 39.76</gml:coordinates>
                </gml:Box>
            </ogc:Within>
        </ogc:Filter>
    </wfs:Query>
</wfs:GetFeatureWithLock>
GetFeatureWithLock response

```
<wfs:FeatureCollection lockId="2567" xmlns:wfs="http://www.opengis.net/wfs" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <gml:boundedBy xmlns:gml="http://www.opengis.net/gml">
    <gml:Box srsName="SDO:8307">
      <gml:coordinates>-104.759899,38.8632 -104.729772,39.712267</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember xmlns:gml="http://www.opengis.net/gml">
    <scottns:UsCities fid="54" xmlns:scottns="http://www.myserver.com/scott">
      <scottns:ID>54</scottns:ID>
      <scottns:CITY>Colorado Springs</scottns:CITY>
      <scottns:STATE_ABRV>CO</scottns:STATE_ABRV>
      <scottns:POP90>281140</scottns:POP90>
      <scottns:RANK90>54</scottns:RANK90>
      <scottns:LOCATION>
        <gml:Point srsName="SDO:8307" xmlns:gml="http://www.opengis.net/gml">
          <gml:coordinates decimal="." cs=""," ts="">-104.759899,38.8632</gml:coordinates>
        </gml:Point>
      </scottns:LOCATION>
    </scottns:UsCities>
  </gml:featureMember>
...
</wfs:FeatureCollection>
```
LockFeature request

```xml
<wfs:LockFeature
    service="WFS"
    version="1.0.0"
    expiry="5"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:scottns="http://www.myserver.com/scott">
    <wfs:Lock typeName="scottns:UsCities">
        <ogc:Filter>
            <ogc:Within>
                <ogc:PropertyName>scottns:LOCATION</ogc:PropertyName>
                <gml:Box srsName="SDO:8307">
                    <gml:coordinates>-105.11, 38.86 -104.72, 39.76</gml:coordinates>
                </gml:Box>
            </ogc:Within>
        </ogc:Filter>
    </wfs:Lock>
</wfs:LockFeature>
```

- Lock duration (in minutes)
- Features to lock selected using the OGC query syntax
LockFeature response

- Response contains the unique, system-generated id for the lock used

```xml
<WFS_LockFeatureResponse
    xmlns="http://www.opengis.net/wfs"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.0.0/
    WFSTransaction.xsd">
    <LockId>2568</LockId>
</WFS_LockFeatureResponse>
```
Locking failure

- Returned when unable to lock one of more features
- Usually because they are already locked

```xml
<ogc:ServiceExceptionReport version="1.2.0"
  <ogc:ServiceException code="WFS-1027">
    Message: Requested Feature instances cannot be locked
    Original exception:
    java.sql.SQLException: ORA-13199: Row cannot be locked
    ORA-06512: at "MDSYS.MD", line 1723
    ORA-06512: at "MDSYS.MDERR", line 17
    ORA-06512: at "MDSYS.SDO_WFS_LOCK", line 215
    ORA-06512: at "MDSYS.SDO_WFS_LOCK", line 86
    ORA-06512: at line 1
  </ogc:ServiceException>
</ogc:ServiceExceptionReport>
```
Transaction request

- A single request can group multiple individual updates (inserts, updates, deletes)
- Executed as a single atomic transaction
- Automatically committed
<wfs:Transaction service="WFS" version="1.0.0"
xmlns:gml="http://www.opengis.net/gml" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:scottns="http://www.myserver.com/scott">
  <wfs:Insert handle="INSERT-01">
    <scottns:UsCities xmlns:scottns="http://www.myserver.com/scott">
      <scottns:ID>196</scottns:ID>
      <scottns:CITY>Bismarck</scottns:CITY>
      <scottns:STATE_ABRV>ND</scottns:STATE_ABRV>
      <scottns:POP90>280000</scottns:POP90>
      <scottns:RANK90>212</scottns:RANK90>
      <scottns:LOCATION>
        <gml:Point srsName="SDO:8307"
          xmlns:gml="http://www.opengis.net/gml">
          <gml:coordinates decimal="." cs=""," ts=" ">
            -100.74869 46.7666667
          </gml:coordinates>
        </gml:Point>
      </scottns:LOCATION>
    </scottns:UsCities>
  </wfs:Insert>
</wfs:Transaction>
<wfs:Transaction
    service="WFS"
    version="1.0.0"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:scottns="http://www.myserver.com/scott">
    <wfs:LockId>2568</wfs:LockId>
    <wfs:Update handle="UPDATE-01" typeName="scottns:UsCities">
        <wfs:Property>
            <wfs:Name>scottns:POP90</wfs:Name>
            <wfs:Value>30000</wfs:Value>
        </wfs:Property>
        <ogc:Filter>
            <ogc:PropertyIsGreaterThan>
                <ogc:PropertyName>scottns:RANK90</ogc:PropertyName>
                <ogc:Literal>200</ogc:Literal>
            </ogc:PropertyIsGreaterThan>
        </ogc:Filter>
    </wfs:Update>
</wfs:Transaction>
Transaction Delete

```xml
<wfs:Transaction
    service="WFS"
    version="1.0.0"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:scottns="http://www.myserver.com/scott">
    <LockId>2568</LockId>
    <wfs:Delete handle="DELETE-01" typeName="scottns:UsCities">
        <ogc:Filter>
            <ogc:PropertyIsGreaterThan>
                <ogc:PropertyName>scottns:RANK90</ogc:PropertyName>
                <ogc:Literal>200</ogc:Literal>
            </ogc:PropertyIsGreaterThan>
        </ogc:Filter>
    </wfs:Delete>
</wfs:Transaction>
```
**Transaction response**

- Contains the ids of the new features inserted (if any)

```xml
<wfs:WFS_TransactionResponse version="1.0.0">
  <wfs:InsertResult handle="INSERT-01">
    <ogc:FeatureId fid="196"/>
  </wfs:InsertResult>
  <wfs:TransactionResult>
    <wfs:Status>
      <wfs:SUCCESS/>
    </wfs:Status>
  </wfs:TransactionResult>
</wfs:WFS_TransactionResponse>
```
Transaction errors

• For example: unique constraint violation

```xml
<ogc:ServiceExceptionReport version="1.2.0"
   xsi:schemaLocation="http://www.opengis.net/ogc
   http://localhost:7001/examples/servlet/xsds/OGC-exception.xsd">
  <ogc:ServiceException code="WFS-1038">
    Message: Exception during processing DB content
    Original exception:
      java.sql.SQLException:
        ORA-00001: unique constraint (SCOTT.US_CITIES_PK) violated
  </ogc:ServiceException>
</ogc:ServiceExceptionReport>
```
Transaction errors

- Trying to update a feature with an invalid lock id

```xml
<ogc:ServiceExceptionReport version="1.2.0"
  <ogc:ServiceException code="WFS-1028">
    Message: Exception occurred during locking feature instances
    Original exception:
      java.sql.SQLException: ORA-13199: Cannot lock exception
      ORA-06512: at "MDSYS.MD", line 1723
      ORA-06512: at "MDSYS.MDERR", line 17
      ORA-06512: at "MDSYS.SDO_WFS_LOCK", line 396
      ORA-06512: at line 1
  </ogc:ServiceException>
</ogc:ServiceExceptionReport>
```

- The lock may have expired
Transaction errors

- Trying to update a feature locked by someone else

```xml
<ogc:ServiceExceptionReport version="1.2.0"
  xsi:schemaLocation="http://www.opengis.net/ogc
  http://localhost:7001/examples/servlet/xsds/OGC-exception.xsd">
  <ogc:ServiceException code="WFS-1028">
    Message:Exception occurred during locking feature instances
    Original exception:
    java.sql.SQLException: ORA-13199: Row cannot be locked
    ORA-06512: at "MDSYS.MD", line 1723
    ORA-06512: at "MDSYS.MDERR", line 17
    ORA-06512: at "MDSYS.SDO_WFS_LOCK", line 215
    ORA-06512: at "MDSYS.SDO_WFS_LOCK", line 86
    ORA-06512: at line 1
  </ogc:ServiceException>
</ogc:ServiceExceptionReport>
```
Concurrent Updating in SQL

• Once a table is registered for updating by WFS it can no longer be updated in regular SQL

```
SQL> delete from us_cities where id = 42;
```

ERROR at line 1:
ORA-13199: Table not locked
ORA-06512: at "MDSYS.MD", line 1723
ORA-06512: at "MDSYS.MDERR", line 17
ORA-06512: at "SCOTT.US_CITIES_CDL", line 1
ORA-04088: error during execution of trigger 'SCOTT.US_CITIES_CDL'

Concurrent Updating in SQL

• Need to specifically enable database transactions on WFS tables

  SQL> exec SDO_WFS_LOCK.EnableDBTxns();

• Then perform the updates

  SQL> delete from us_cities where id = 42;

• The update will fail if the WFS has a lock on one or more of the rows being updated.
What are the current locks?

- Check the locking tables
- Active locks, their expiry time, and the number of rows locked for each table

```sql
select t.tokenid lockid, t.expirytime, r.tablename, count(*) numrows
from  mdsys.tokensessionmap_t$ t,
     mdsys.rowtokenmap_t$ r
where t.expirytime >= current_timestamp
and    t.tokenid = r.tokenid
group by t.tokenid, t.expirytime, r.tablename;
```

<table>
<thead>
<tr>
<th>LOCKID</th>
<th>EXPIRYTIME</th>
<th>TABLENAME</th>
<th>NUMROWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>104</td>
<td>05-MAR-09 07.51.33.000000 PM</td>
<td>SCOTT.US_CITIES</td>
<td>1</td>
</tr>
<tr>
<td>105</td>
<td>05-MAR-09 07.51.50.000000 PM</td>
<td>SCOTT.US_CITIES</td>
<td>2</td>
</tr>
</tbody>
</table>

2 rows selected.
What features are locked?

- Active locks, their expiry time, and the rows locked for each table

```
select t.tokenid lockid, t.expirytime, r.tablename, r.rowid
from mdsys.tokensessionmap_t$ t,
     mdsys.rowtokenmap_t$ r
where t.expirytime >= current_timestamp
and   t.tokenid = r.tokenid
order by t.tokenid, t.expirytime, r.tablename;
```

<table>
<thead>
<tr>
<th>LOCKID</th>
<th>EXPIRYTIME</th>
<th>TABLENAME</th>
<th>ROWID</th>
</tr>
</thead>
<tbody>
<tr>
<td>104</td>
<td>05-MAR-09 07.51.33</td>
<td>SCOTT.US_CITIES</td>
<td>AAAPdOAACAAAHnuAAD</td>
</tr>
<tr>
<td>105</td>
<td>05-MAR-09 07.51.50</td>
<td>SCOTT.US_CITIES</td>
<td>AAAPdOAACAAAHnuAAE</td>
</tr>
<tr>
<td>105</td>
<td>05-MAR-09 07.51.50</td>
<td>SCOTT.US_CITIES</td>
<td>AAAPdOAACAAAHnuAAF</td>
</tr>
</tbody>
</table>

3 rows selected.
Caching

- Caching entire tables
- Specify tables to be cached in WSConfig.xml file

```xml
<cached_feature_types>
  <feature_type ns="http://www.myserver.com/scott"
    name="UsCities" />
  <feature_type ns="http://www.myserver.com/scott"
    name="UsCounties" />
</cached_feature_types>
```

- We still always query database
  - Only record “ids” returned
  - This is the column specified as “pkeyCol” when publishing the table
  - Record “ids” then matched with rows in cache
Caching

- Cache refreshed from database at regular intervals
- Interval set in WSConfig.xml file (in milliseconds)
  <wfs_cache_sync_interval>60000</wfs_cache_sync_interval>
- Default interval is 10000 (10 seconds)
- Changed rows are recorded in “change table” mdsys.WFS_featureTableUpdated$
- Applications must call
  SDO_WFS_PROCESS.InsertFtDataUpdated() to record updated rows in the change table
Mapviewer Access to WMS and WFS
Using WMS and WFS in MapViewer

Application
JavaScript Map API

Map Cache Server
FOI Server

Map Rendering Engine

HTTP

WMS
WFS

Spatial Tables
Map Definitions

Mapbuilder
Define a **WMS Theme**

URL to get the capabilities document
Define a WMS Theme

Choose layers

Choose coordinate system
Define a WMS Theme

Choose image format, background, transparency
Define a Map over the WMS theme
Define a Cache on the Base Map

Tile Layer Definition

```
<map_tile_layer name="WMS US_BASE_MAP" image_format="PIIIG" http_header_expires="168.0"
    concurrent_fetching_threads="3">
    <internal_map_source data_source="SCOTT" base_map="WMS US_BASE_MAP" bgcolor="#ffffff"/>
    <tile_storage root_path="/tmp"/>
    <coordinate_system srid="8307" minX="-180.0" minY="-90.0" maxX="180.0" maxY="90.0"/>
    <tile_image width="256" height="256"/>
    <zoom_levels levels="20" min_scale="645309.0" max_scale="2.13103147E8">
        <zoom_level tile_width="128.0" tile_height="128.0" scale="2.13103147E8"/>
        <zoom_level tile_width="64.0" tile_height="64.0" scale="1.57043612E8"/>
        <zoom_level tile_width="32.0" tile_height="32.0" scale="1.15731262E8"/>
        <zoom_level tile_width="16.0" tile_height="16.0" scale="7.692307692307693"/>
        <zoom_level tile_width="8.0" tile_height="8.0" scale="4.631716E7"/>
        <zoom_level tile_width="4.0" tile_height="4.0" scale="3.413283E7"/>
        <zoom_level tile_width="2.0" tile_height="2.0" scale="2.5153751E7"/>
    </zoom_levels>
</map_tile_layer>
```

MapViewer Server: http://localhost:7001/mapviewer
MapViewer Data Source: scott
Center X: 0
Center Y: 0
SRID: 8,307
Zoom Level: 9
Define a **WFS Theme**

URL to get the capabilities document
Define a WFS Theme

Choose feature type
Choose spatial column
Choose attributes
Using WMS/WFS Themes in Applications

• Use them like any other theme in your map definitions
• Can mix and match regular themes and WFS themes on the same base map
• Can also use WMS/WFS themes in map requests
• Can use WMS themes for base maps in Oracle Maps
• Can use WFS themes as FOI theme for Oracle Maps
Other Web Services

- Catalog Service for Web (CSW)

- Location Services (OpenLS)
  - Geocoding Service
  - Routing Service
  - Mapping Service
  - Directory Service