REST Service Publishing Guidelines at Iowa DOT

This document provides a framework for creation of Shared or Authoritative REST services.

Table of Contents

Roles

Data Owner: The person that owns the data in the service.
Data Subject Matter Expert (Data SME): A person knowledgeable about the data being discussed. This can be the same person as a Data Owner.
REST Subject Matter Expert (REST SME): A person knowledgeable about the requirements and standards set forth below and is responsible for working with the Data Owner/Data SME to create the FGDB, APRX file, and publish the REST service. Usually this person is within the same division as the Data Owner/Data SME but is not the Data Owner or Data SME.
GIS Admin: A representative from the enterprise geospatial support team that has access to appropriate SDE admin user accounts to create feature classes, domains, and tables.
IT Admin: A representative from IT that supports the Data Owner's division.
Application Owner: An individual or end client that who owns/develops an application that utilizes the service(s) being discussed.

Process

1. Initial Discussion (see Page 3)
2. Test Feature Class(es) Created
    a. The Data Owner, Data SME, and REST SME will work together to develop a FGDB, ensuring the Minimum **Data Requirements** are met. It should include a complete dataset for publishing with all attributes and geometry.
    b. Once all requirements are met AND IF the selected database option is the Relational Data Store, the REST SME will provide the GIS Admin the FGDB. The GIS Admin will create new feature class(es) and table(s) using the FGDB on a Test Database (SDE or Relational Data Store) and provide connection details to the REST SME.
3. Test APRX Created/Reviewed + REST Service(s) Published
    a. REST SME will build APRX file following the **APRX Requirements** and publish to the appropriate Test ArcGIS Enterprise Environment or AGOL.
4. Test User Acceptance Testing of Test REST Service
    a. Appropriate personnel (including but not limited to the REST SME) will adequately review and test the service ensuring all the requirements have been met and it functions correctly. The REST SME and the GIS Admin will reach out and notify Application Owners, who will need to verify their Web/Desktop Application(s) account for changes to an existing service or to add a new service. **No further tasks will be completed until UAT has completed and the <u>REST SME + GIS Admin</u> verifies the new service or changes are acceptable.**
5. Production Feature Class(es) Created
    a. Once all testing is completed, the GIS REST Admin will create the new feature class(es) on a Production Database (SDE or Relational Data Store).
    b. A new APRX file will be recreated with the new Production Database connections
6. Production REST Service(s) Published
    a. GIS REST Admin will publish new production REST Service
    b. If modifying an existing service, a set upgrade time window will be determined to limit downtime for production applications.
7. Final User Acceptance Testing
    a. Appropriate personnel (including but not limited to the REST SME) will adequately review and verify the service ensuring all the requirements have been met and it functions correctly. The REST SME and the GIS Admin will reach out and notify Application Owners, who will need to verify their Web/Desktop Application(s) account for changes to an existing service or to add a new service. **Once the Final UAT has concluded, no further changes can be made unless the process starts over.**
8. Update Continuity Documentation
    a. In the future, an enterprise, searchable database will be maintained for all REST services and related applications that are powered by all REST services.

Initial Discussion

When a new REST service is needed or a significant change to existing REST service is needed

Convene a meeting with the following individuals to discuss the service:

1. Data Owner(s)
2. Subject Matter Expert(s) (Data and REST SMEs)
3. Appropriate GIS Admin(s)
4. Appropriate IT Admin(s)
5. Appropriate Application Owner(s) - At the very minimum they should be notified that the service is going to be changing

At the end of this discussion, the following points should be answered:

- Who is the Data Owner, Data SME, and REST SME?
- What is the Purpose of the REST Service?
- Does the new service fulfill the purpose - OR - Do the proposed change(s) improve usability and accuracy while fulfilling the purpose?
- Where/Who is the service going to be used? (Applications? Third Party Applications?)
- When is the service going to primarily be used? (Business Hours? 24/7?)
- How is the data updated? (FME, etc.?)
- How often is the data updated?
- How often is REST service updated?

    Example: 5-year plan REST service changes every year
- Who will have access to the service? (Public, DOT or group) - If DOT-Only, need to justify why it can't be public.
- Does there need to be a transactional service and a publication service?
- Is service authoritative (SOR) or non-authoritative (SOE)?
- Where the service will be published (SOE, SOR, External, Analytics/Real-Time, or AGOL)
- Determine best location for where data will reside, understanding the limitations of each:

    SDE/Oracle/SQL Server should be used when other business units/systems need to interact with the data or the data already exists in a database. Keeping your data in SDE will allow you to use a DBMS to run SQL queries against the data. The only drawback is that if you are not able to easily create "Hosted Feature Views" from your services, instead you will need to publish multiple APRX files with SQL queries if you wish to recreate "view based" services.

    Relational Data Store (Hosted Feature Layer) should be used when data is only needed to be used in GIS systems. Data can only be queried through REST services. Hosted Feature Views can be easily spun up.
- Timeline for the following tasks:
    1. Test Feature Class Creation
    2. Test APRX Creation
    3. Text APRX Review
    4. Test REST Service Published
    5. User Acceptance Testing (Update/Test Affected Test Web Applications)
    6. Production Feature Class Creation
    7. Production APRX Creation
    8. Production APRX Final Review
    9. Production REST Service Published

Minimum Data Requirements

Quick Checklist:

- ✔ **Attribute(s)**, **Domain(s)**, and **SubType(s)** meet requirements stated below (REQUIRED)
- ✔ **Geometry** meets requirements stated below (REQUIRED)
- ✔ **Feature Class(es)** and **Table(s)** are named appropriately (REQUIRED)
- ✔ **Metadata** is filled out and meets the minimum requirements stated below (REQUIRED)
- ✔ No joins or related tables should be included for REST service publication except where appropriate. GIS REST Admin will provide guidance.

Attribute(s), Domain(s), and SubType(s):

- ❑ Data Field Types are appropriate
- ❑ Human-readable field aliases
- ❑ Domain values are used where appropriate
- ❑ Domains should utilize existing domains already in place where possible
- ❑ Domains should be used more than once where applicable
- ❑ Coded Values and Descriptions should not be ‘just duplicated’
- ❑ Subtypes are setup correctly and used when appropriate
- ❑ Attributes should follow best practices and the Master Data Management Plan standards

Geometry:

- ❑ X,Y,Z,M Tolerance is set to 0.000005
- ❑ All feature classes are ‘Z-enabled’ (elevation) whether there’s elevation data to be entered or not
- ❑ Polyline geometry type feature classes should be ‘M-enabled’ (direction) for future use
- ❑ SHAPE.AREA field has an alias of “Area (SqM)”
- ❑ SHAPE.LENGTH field has an alias of “Length (Meter)” for lines. For polygons: set the alias to “Perimeter (Meter)”
- ❑ All data is in LRS Lambert OR WGS 1984 Web Mercator Auxiliary Sphere (3857)
- ❑ Service extents for statewide data

  Left: -96.947 Right: -89.981 Top: 43.751 Bottom: 40.259

Metadata Minimum Requirements:

- ❑ ISO 19139 Metadata Standard is used
- ❑ Overview Section
  - ○ “Summary” is filled out
  - ○ “Description” is filled out
  - ○ “Credits” is filled out

Example: Iowa Department of Transportation – Traffic & Safety

- “Tags” filled out

  In the future, a ‘Tag’ library will be created to choose from.

- “Use Limitations” filled out

□ Resources Section

- “Points of Contact” is filled out with only the office name of the data owner. There should not be any individual names/contact information listed.
- “Maintenance” – filled out with update frequency of the data
- “Fields” – A definition of each field must be filled out

□ Metadata does not reference any internal database, database schema name, database table name/view, file, path, server name, IP address, etc.

□ Try to add metadata to the feature class used for publishing the service. This will carry through into the .aprx file and published REST service.

SDE/Oracle/SQL Requirements

- □ If data is not confidential: *GRANT SELECT ON* ***Your_Table_Name*** *TO GIS_S_ROLE;*
- □ When bulk loading large data sets it is best to drop the spatial index, load table, then recreate the spatial index.
- □ SDE registration is optional and is only needed if data needs to be written to by ESRI products. Use of Query layers allows direct connection to Oracle without SDE.
- □ The table contains no columns of a user-defined type. If a table has user-defined types, then a view can be made without the user-defined types.
- □ Each table must have an OBJECTID, that is a SINGLE, UNIQUE, NOT-NULL INTEGER data-type column that may be used as ROW_ID column for ArcSDE. If the column is NUMBER(38) NOT NULL UNIQUE or NUMBER(38) PRIMARY KEY, it will be automatically registered as the row ID with ArcSDE, but this isn't necessary.
- □ Mixed-case table and column names are not allowed. All values representing database object must be in uppercase and must conform to the standard Oracle conventions for table names. Oracle has many reserved words and these must not be used for table or column names. Using reserved words can have unpredictable results.
- □ Each layer in the APRX file must contain a single type of geometry. (That is, all geometries in the geometry column should have the same SDO_GTYPE). Geometry may be multipart. Views used to publish data must limit the geometry types because of ESRI constraints. For example a view that contains point data must be filtered to only allow points and point collections *SELECT GEOMETRY FROM* ***YOUR_SCHEMA_NAME****.****YOUR_TABLE_NAME*** *DATA WHERE DATA.GEOMETRY.sdo_gtype = 2001;*

- □

If using SDO Geometry:

- □ Each table must have a single SDO_GEOMETRY column. If more than 1 geometry column exists, a view will need to be created that references only one.
- □ Each table must have a valid entry in MDSYS.USER_SDO_GEOM_METADATA. Spatial indexes require the presence of Oracle's spatial metadata. See appendices on how to register metadata.
- □ Each table must have a spatial index. See appendices on how to register
- □ Each geometry type must be supported by ArcSDE. Unsupported geometry types include: Application-specific geographic elements (SDO_ETYPE 0) Optimized rectangles (SDO_ETYPE 3, interpretation 3) Optimized circles (SDO_ETYPE 3, interpretation 4) Heterogeneous collections (SDO_GTYPE D004)

APRX Requirements

Quick Checklist

- Symbology
- Metadata

Detailed

- Uses authoritative symbology. APRX files should NOT be customized for specific client applications, as each client should symbolize the service based on their needs. Contact Cartography if you have questions or require symbology for your aprx file.
- Use OS Authentication as much as possible. This prevents services from breaking when passwords get changed.
- Note: Using Database Authentication with a non-changing username\password is also acceptable.
- Pull data from warehouses when possible for publication REST services instead of from transactional systems. If you are publishing a service meant for editing, you should connect directly to the transactional system.
- When an APRX file has layers for each year, the top most layer should be the most current year, followed by the oldest year and ending with the most current year. The purpose behind this is to ensure that applications won’t break when adding new years, and they have a consistent method to access the most current year of data. Example: Layer 0: Current Year, Layer 1: 2015, Layer 2: 2016 ... Layer 7: 2021

Metadata

- □ Verify metadata is completely brought in for each layer. If some metadata is missing, populate it appropriately.
  - Title
  - Tags
  - Summary
  - Description
  - Credits
  - Use limitations

- □ Add metadata for the map
  - Title
  - Tags
  - Summary
  - Description
  - Credits
  - Use limitations

PORTAL Group Creation

Portal group management will be managed by GIS team along with IT. Goal is to allow consistent group creation and allow for Azure group rules to manage Portal groups.

There will three user roles used in portal

- Standard User – Ability to create content and be added to groups. User cannot share content to the organization. This is the default user
- Divisional User – Set to a service account to manage groups and content
- Administration User – Administrator of the system

Workflow for group Creation

1. IT or GIS sends a request to create new Azure AD Group
2. When Azure AD group is created, GIS team is notified of new group and will create a linked enterprise group inside portal. GIS team will need the Azure ObjectID of the group, Divisional user name to manage the group and desired group name. Note: Azure and portal groups will have a one to one relationship.
3. GIS will inform group owner that the group has been created.

Publishing

System of Engagement (SoE): A System of Engagement (SoE) manages and promotes user collaboration and interaction. It overlays and complements an organization's investment in a system of record by providing easy access to data as well as easy to use applications that enable collaboration across your organization (ESRI). Examples of Iowa DOT SoE services are Maintenance Culverts or Survey Markers.

System of Record (SoR): A System of Record is data used to collect, manage, and compile geospatial information in maps, which represent reality in a lucid way and allow for better analysis and more sound decision making (ESRI). Examples of Iowa DOT SoR services are City Boundaries or Statewide Parcels.

Appendix

Oracle Spatial 11 spatial guidelines

Steps to create geometry in an Oracle Database

1. Determine tables to store geometry
   a. Create field called GEOMETRY of type SDO_GEOMETRY
2. Create spatial metadata using the appropriate SRID.
   a. Example using Lambert Conformal Conic: *INSERT INTO USER_SDO_GEOM_METADATA values('**YOUR_TABLE_NAME**', '**GEOMETRY_COLUMN_NAME**', MDSYS.SDO_DIM_ARRAY(MDSYS.SDO_DIM_ELEMENT('X', **-2147483648**, **2147483647**, 0.000005), MDSYS.SDO_DIM_ELEMENT('Y', **-2147483648**, **2147483647**, 0.000005)), **1050010**); COMMIT;*
3. Create spatial index for geometry column use _SPIDX at the end of index name.
   a. Example: *CREATE INDEX **SCHEMA_NAME.INDEXNAME**_SPIDX ON **SCHEMA_NAME.YOUR_TABLE_NAME**(GEOMETRY) INDEXTYPE IS MDSYS.SPATIAL_INDEX PARAMETERS('sdo_non_leaf_tbl=TRUE');*
   b.
   c. For 11g, if geometry is of type point then use parameter LAYER_GTYPE=POINT
   d. If your table has a large amount of geometry, it is recommended to use the parameter *sdo_non_leaf_tbl=TRUE* when creating the index, then pinning the related MDNT table to RAM.
   e. You can get the non leaf table name by running the following query: *SELECT sdo_nl_index_table FROM user_sdo_index_metadata WHERE sdo_index_name='**YOUR_INDEX_NAME**_SPIDX';*
   f. Pin table to memory: *ALTER TABLE MDNT_INDEX_TABLE_QUERY$ STORAGE(BUFFER_POOL KEEP);*

Tips

- If bulk loading large data sets it is best to drop the spatial index, load table then recreate spatial index.
- DBMS stats are OK to run on spatial index tables but do not offer improvement in performance
- Good practice to analyze the RTREE indexes for indexes that need to be rebuilt

    - Example: *SET SERVEROUTPUT ON SIZE 20000 SET TIMING ON DECLARE CURSOR c_1 IS SELECT U.SDO_INDEX_OWNER, U.SDO_INDEX_NAME FROM MDSYS.USER_SDO_INDEX_METADATA U; BEGIN DBMS_OUTPUT.PUT_LINE ('Begin index check'); FOR c1_rec IN c_1 LOOP BEGIN SDO_TUNE.ANALYZE_RTREE (c1_rec.SDO_INDEX_OWNER, c1_rec.SDO_INDEX_NAME); END; END LOOP; END; /*

Requirements for working with Oracle Spatial 11

- Only one set of attribute data is allowed for any given geometry. The use of attribute tables in a join-view relationship with a table containing geometry information allows you to get around this limitation.
- All spatial filter operations are performed on the Oracle Server, which requires spatial indexes to exist for all feature classes. This greatly improves spatial filter performance for filter areas that are less than 70% of the total area covered by the feature class.
- In order to be editable, all tables must have a primary key. Primary keys must be numeric or alphanumeric. Integer-based primary keys, populated by an associated sequence, are recommended and will provide the best results.
- For views, a primary key is required for at least one of the tables that will be used in the view definition. This is known as a key preserved view.
- The Oracle Object Model data server supports the use of sequences for each field that needs to be treated as AutoNumber. The most common use of AutoNumber is primary key fields.
- When creating triggers to manage a sequence for the primary key, the trigger must be similar to the code below to allow other software to manage the primary key values via a sequence but not stop other applications from adding records. You can assign sequence for use with any numeric field, but only one field can have the distinction of being autonumber.
    - *CREATE OR REPLACE TRIGGER STATES_INSERT_TRIG BEFORE INSERT ON* ***YOUR_TABLE_NAME*** *REFERENCING OLD AS OLD NEW AS NEW FOR EACH ROW IF :NEW.****YOUR_AUTONUMBER_ATTRIBUTE*** *IS NULL THEN SELECT* ***YOUR_TABLE_NAME_YOUR_AUTONUMBER_ATTRIBUTE****_SEQ.NextVal INTO :NEW.****YOUR_AUTONUMBER_ATTRIBUTE*** *FROM dual; END IF; END;*

Other Guidelines

- For you to create a view on a table that is owned by another user, the table owner must grant you SELECT privileges with the GRANT option on the table (or tables) you want to use in the view. *grant select on view_dept_100 to mgr100 with grant option;*
- It should pass Oracle’s geometry validation tests; otherwise, accessing these geometries may have unexpected results. Additional information on how Oracle validates geometry can be found at the following link: http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14255/sdo_objgeom.htm#BGHFDDBF.
- Geometries in the table should use the LRS Lambert conformal conic projection. This projection is represented in Oracle Spatial by the SDO_SRID value of 1050010.
- Certain Oracle Spatial geometry types are unsupported by ArcSDE, so they will not work with GeoNexus:
    - Supported:
        - Simple point (SDO_GTYPE D001)
        - Simple line string (SDO_GTYPE D002)
        - Simple polygon (SDO_GTYPE D003)
        - Multipoint (SDO_GTYPE D005)

    - Multiline (SDO_GTYPE D006) – Note: Multiline geometries sometimes give unpredictable results, so they should be avoided if possible.
    - Multipolygon (SDO_GTYPE D007)
    - Compound line strings (SDO_GTYPE D002, SDO_ETYPE 4)
    - Compound polygons (SDO_GTYPE D003, SDO_ETYPE 1005 or 2005)
- Unsupported:
    - Application-specific geographic elements (SDO_ETYPE 0)
    - Optimized rectangles (SDO_ETYPE 3, interpretation 3)
    - Optimized circles (SDO_ETYPE 3, interpretation 4)
    - Heterogeneous collections (SDO_GTYPE D004)