##### Define new information exchange

*Please also see* [*https://github.com/usgin/usginspecs/wiki/Define-New-Information-Exchange*](https://github.com/usgin/usginspecs/wiki/Define-New-Information-Exchange) *which is the ‘living document’ that will be revised and updated as practice evolves. For an outline regarding content model creation specifications, please see* [*https://github.com/usgin/usginspecs/wiki/Content-Model-Guidelines*](https://github.com/usgin/usginspecs/wiki/Content-Model-Guidelines)*.*

The decision to define a new information exchange should be based on the likelihood that others will want to publish similar datasets in the future. Members of the USGIN community propose specifications for data sharing exchanges. Exchange documents are developed and reviewed using a publicly accessible repository on GitHub (<https://github.com/usgin-models>). Each exchange has a separate repository associated with the usgin-models pseudo organization. A proposed model must have an identified steward, and a working group of at least three participants with relevant domain knowledge and understanding of the interchange technology. There is no formal process for defining workgroup membership; normally the challenge is finding a sufficient number of qualified individuals to provide meaningful reviews and comment. The exchange steward is responsible to assemble the workgroup and assure sufficient expertise in the group to generate a sound content model and implementation. The exchange steward requests creation of a new model repository at the usgin-models gitHub from the organization members, and identifies workgroup members who will have commit privleges on the repository. Any community member can create a repository branch to propose changes using standard GitHub procedures, and request consideration for merging back into the developing model.

After review and approval by the workgroup, a call goes out to a USGIN technical review e-mail list or by RSS feed (<http://notifications.usgin.org/>) for comments from the community. An open review period of 4 weeks is normal, after which any comments from the community must be resolved to the satisfaction of the commenter. When issues are resolved to the satisfaction of the stakeholders (workgroup and engaged community), the exchange specification is adopted.

When a specification is adopted, all associated documents are copied to a 'tag' branch in the gitHub repository and are not changed after they are 'tagged'. The Specification documents are also copied to the exchange repository at <http://schemas.usgin.org>, which is a web site set up to provide public access to exchange specifications and any related xml schema documents or other artifacts required for the deployment and operation of the information exchange.

Here is a more detailed outline for setting up a USGIN information exchange:

1. Steward assembles workgroup, defines scope of model, and gets repository set up on usgin-models gitHub at <https://github.com/usgin-models>.
2. Workgroup defines content model. The recommended procedure is to scope the model based on stated target use scenarios, and on example datasets that the interested parties want to share. When creating a content model draft, please see ‘ContentModelGuidelines’ document at https://github.com/usgin/usginspecs. To gather interested individuals, the USGIN Notifications system via RSS feed may be used at <https://github.com/ngds/service-notifications/issues> where new a new issue is posted to introduce the proposal.
3. Get review of workgroup draft from community of expected users. Revise content model as necessary and post drafts in the GitHub repository.
4. Select interchange service protocol. This should be based on the availability of server and client software in the community of users. Some example possibilities include OGC WFS, WMS, WCS, OpenDAP/NetCDF/THREDDS, Microsoft OData, ESRI Geoservices API.
5. Implement the content model using an encoding scheme compatible with the chosen service protocol, e.g. XML, JSON, turtle, csv.
6. Define validation rules for instance documents
7. Document the content model, interchange format, service protocol, and any special conventions or profile. Specify how data access links to data exposed using this exchange will be described using the metadata fields in USGIN profile ISO19139 CI\_OnlineResource elements. Use existing identifiers where possible to identify service and MIME types.
8. Deploy an example service and test with client software. Iterate 1-5. Register a dataset distributed using the exchange in a USGIN catalog.
9. Have documentation reviewed by target users and technical experts; respond to comments, updating 1-7 as necessary. If using USGIN Notifications, post comments on the open issue to alert the working group and/or community to ensure all parties have commented.
10. Adopt content model for implementation. Tag the version of the content model and specification documents in the GitHub repository, and deposit specification documents in USGIN exchange repository at <http://schemas.usgin.org/models> at which time it is made available to the public.

Outline for creating a new version (editing an existing content model):

1. Any member of working group makes edits to content model in the given repository at <https://github.com/usgin-models>. Any interested party may also make a new branch and make edits, but will need appropriate privileges from the steward of that repository to push changes to the master branch for the community to view.
2. Notifications regarding changes in the content model must be posted at <https://github.com/ngds/service-notifications/issues> to be read by the RSS feed at <http://notifications.usgin.org/>. This issue must indicate reasons for the change and a reasonable comment period.
3. Comments and edits continue in the GitHub repository.
4. Following approval by the working group and the steward, a new version of the content model is indicated on the “About” tab of the content model Excel file and tagged in the GitHub repository, rendering the previous version of the content model deprecated.
5. The new version of the content model is then uploaded by an administrator at <http://schemas.usgin.org/models/> to be made available to the public.

### Checklist for exchange steward:

Before exchange specification documents are tagged in GitHub and put online at schemas.usgin.org/models, the following checks should be made:

* Field headings are consistent in spelling, capitalization, and order in all tabs of the Excel workbook (Data and FieldList tabs).
* Field headings spelling, capitalization, order, and cardinality are exactly the same in the Excel workbook (as described in the FieldList tab) and in the schema (XSD).
* The schema (XSD) has the first field as “OBEJCTID”.
* The schema (XSD) has the last field as “Shape”.  Where both “ShapeLength” and “Shape” fields are used, “ShapeLength” is second to last, and “Shape” is the last field. [xs:any, ShapeArea]

Repository management:

A good first step is to download the GitHub GUI at <https://github.com/> where you will click “Set up Git” and follow the steps. Be sure to always create a “readme.md” file to explain the purpose of the information exchange, indicate the content in the given repository, and provide any updates regarding the version in use or other pertinent information.

Create a new repository by going to the USGIN GitHub site at <https://github.com/usgin-models> and logging into GitHub. Next to the user name in the upper right-hand corner, click “+” and choose “New repository”. Once created, click the “Clone in Desktop” button in the repository to get a local copy and begin entering documents into your local folder. Once ready to push the changes to the “master” to be viewed by others, commit the changes and “sync” in the GUI. Edit existing content models in much the same way; clone the repository in your desktop then edit and sync your changes back to the master branch.

File naming conventions:

The name of the schema (XSD) and Excel (XLS) represenations of the information exchange should be identical and reflect the exact name of the eventual service.