



ArcGIS for INSPIRE 10.8.2 Server Extension Customization Guide on INSPIRE View Service Layer

Content

1	Introduction.....	1
2	Introduction to LayerInfo Table	1
2.1	Purpose.....	1
2.2	Table Structure Explained	2
2.2.1	List of Fields	2
2.2.2	Example: Protected Sites—Archaeological	3
2.3	Use Case: Add Custom Layers	4
2.4	Use Case: Add Scale Dependencies.....	6

1 Introduction

The purpose of this document is to provide instructions on configuring and customizing ArcGIS for INSPIRE in terms of an Infrastructure for Spatial Information in Europe (INSPIRE) View service's scale dependencies and layer structure.

Follow these instructions to add custom scale dependencies and default visibilities to each of the system-maintained INSPIRE layers. This document also explains how ArcGIS for INSPIRE sets up the INSPIRE portrayal layer structure.

2 Introduction to LayerInfo Table

The INSPIRE Annex I data model implemented using Esri ArcGIS geodatabase (GDB) technology consists of several tables and feature classes. Most of these tables are directly derived from the INSPIRE UML model, and they are also enriched with other properties to drive the system.

Important: Please note that customized layers may affect the proper functioning of existing MXD documents as well as INSPIRE services.

2.1 Purpose

The data model contains some meta tables that bring further knowledge about INSPIRE into the ArcGIS for INSPIRE software. Concerning INSPIRE View services, the LayerInfo table plays an important role as this table contains information about each layer listed in the Portrayal section of the INSPIRE data specifications.

The table also fills the gap between assumptions INSPIRE makes on the way spatial objects are being exposed through INSPIRE network services and the way Esri's GDB technology works. The most obvious example is the fact that INSPIRE does not distinguish between point, line, and polygon feature types, though the Esri geodatabase does when creating a feature class. Consequently, this needs to be hidden for a consumer of an INSPIRE View service; the basis for this is laid out in the LayerInfo table.

2.2 Table Structure Explained

Each row of the table represents a single layer that can be added to an ArcMap document.

2.2.1 List of Fields

Field Name	Explanation
objectid	GDB internal ID
id	Application internal ID
theme	Name of the INSPIRE theme (e.g., Protected Sites)
fc_name	Name of the feature class on the GDB level, hyphen used if layer is a group layer
stype	Subtype to be used; -1 if none
app_schema	Name of the INSPIRE application schema (e.g., Hydro – Physical Waters)
ir_version	Not used
layer_name	INSPIRE name for the layer or extended name for spatial type
layer_title	Layer title to be used in ArcMap table of contents
spatial_object_type	Name of the INSPIRE spatial object type the layer belongs to
def_query	Definition query to use when creating the layer in ArcMap
parent_id	Reference to parent layer ID (field ID), used for grouping layers; -1 if no parent layer present
spatial_object_type_prefix	Prefix of the INSPIRE spatial object type
layer_keywords	Keywords for the layer
is_hidden	Indicates which layer should be hidden from the View service's user
is_visible	Indicates whether the layer should be visible by default (when adding it to ArcMap using the ArcGIS for INSPIRE Add-in); can be null
min_scale	Minimum scale of the layer (when adding it to ArcMap using the ArcGIS for INSPIRE Add-in); can be null
max_scale	Maximum scale of the layer (when adding it to ArcMap using the ArcGIS for INSPIRE Add-in); can be null
GDBTEMPLATE_NAME	Allowed values: A1, GE, LC, representing respectively Annex I, Geology, and Land Cover

2.2.2 Example: Protected Sites—Archaeological

See the following excerpt from the table LayerInfo (split into two parts for visibility reasons):

Object_ID	ID	Theme	fc_name	type	app_schema	layer_name
300	300	Protected Sites	-	-1	Protected Sites	PS.ProtectedSitesArchaeological
96	96	Protected Sites	psSiteL	-1	Protected Sites	PS.ProtectedSitesArchaeological.L
97	97	Protected Sites	psSiteMP	-1	Protected Sites	PS.ProtectedSitesArchaeological.MP
95	95	Protected Sites	psSiteP	-1	Protected Sites	PS.ProtectedSitesArchaeological.P
94	94	Protected Sites	psSiteS	-1	Protected Sites	PS.ProtectedSitesArchaeological.S

layer_title	spatial_object_type	def_query	parent_id	is_hidden
Protected Sites - Archaeological	-	-	90	0
Protected Sites - Archaeological (Line)	ProtectedSite	protclass = 'Archaeological'	300	1
Protected Sites - Archaeological (Multipoint)	ProtectedSite	protclass = 'Archaeological'	300	1
Protected Sites - Archaeological (Point)	ProtectedSite	protclass = 'Archaeological'	300	1
Protected Sites - Archaeological (Surface)	ProtectedSite	protclass = 'Archaeological'	300	1

There are five layers defined belonging to the theme Protected Sites. The layer holding an ID of 300 is an INSPIRE layer taken from the relevant data specification. The official layer name is PS.ProtectedSitesArchaeological. The four subsequent layers exist due to the differentiation in the Esri geodatabase: point, multipoint, line, and surface are stored in different feature classes (psSiteL, psSiteMP, and so forth). These four layers reference the first layer (parent_id → id). The definition query only allows objects matching a specific protection class. The property is_hidden tells the system that only the first layer should be exposed via the INSPIRE View service (hide the technical details).

2.3 Use Case: Add Custom Layers

If users are going to use ArcGIS for INSPIRE for Protected Sites, it is possible that national legislation introduced specific designations and designation schemes to this domain that are not yet covered by ArcGIS for INSPIRE. Additionally, the data specification is open to changes in which portrayal layers can be offered to the client.

To understand how customization works, users must know that

- Each INSPIRE layer consists (normally) of four hidden sublayers (point, multipoint, line, surface).
- Some feature classes can carry multiple INSPIRE layers (definition query and subtypes).
- An ArcMap group layer bundles the underlying hidden sublayers.
- The group layer will be the INSPIRE layer in terms of the INSPIRE View service.

Returning to the initial example, Protected Sites, there is the following structure:

```
+ Protected Sites
|-- Archaeological                               Group Layer & INSPIRE Layer
|---- Archaeological [Surface]                   hidden Sublayer
|---- Archaeological [Point]
|---- Archaeological [Line]
|---- Archaeological [Multipoint]
|-- Ecological                                   Group Layer & INSPIRE Layer
|---- ...
|-- NationalMonumentsRecord                     Group Layer
|---- AgricultureAndSubsistence                 Group Layer & INSPIRE Layer
|----- AgricultureAndSubsistence [Surface]    hidden Sublayer
|----- AgricultureAndSubsistence [Point]
|----- AgricultureAndSubsistence [Line]
|----- AgricultureAndSubsistence [Multipoint]
|----- ==== new example section with new designation "Desig" in Natura 3000 ====
|-- Natura 3000                                 Group Layer
|---- Desig                                     Group Layer & INSPIRE Layer
|----- Desig [Surface]                         hidden Sublayer
|----- Desig [Point]
|----- Desig [Line]
|----- Desig [Multipoint]
```

To create the six new layers inside ArcGIS for INSPIRE, six SQL statements are needed to put in the new metadata:

1. Create the new group layer Protected Sites - Natura3000 and create a reference to the parent layer Protected Sites (<id0>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer_name, layer_title, spatial_object_type, def_query, parent_id,
spatial_object_type_prefix, layer_keywords, is_hidden, is_visible) VALUES
(<id1>,<id1>,'Protected Sites','-','-1','Protected Sites',0,'PS.ProtectedSitesNatura3000',
'Protected Sites - Natura3000', '-','-','<id0>', '-','conservation,Natura3000', 0, 1);
```

2. Create the new group and INSPIRE layer Protected Sites - Natura3000 – Desig and create a reference to the parent layer created in step 1 (<id1>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer_name, layer_title, spatial_object_type, def_query, parent_id,
spatial_object_type_prefix, layer_keywords, is_hidden, is_visible) VALUES
(<id2>,<id2>,'Protected Sites','-','-1','Protected Sites',0,'PS.ProtectedSitesDesig ',
'Protected Sites - Natura3000 - Desig', '-','-','<id1>', '-','conservation,Natura3000',
0, 1);
```

3. Create the hidden sublayer holding geometries of type Line, use a specific definition query, and create a reference to the parent layer created in step 2 (<id2>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer_name, layer_title, spatial_object_type, def_query, parent_id,
spatial_object_type_prefix, layer_keywords, is_hidden, is_visible) VALUES
(<id3>,<id3>,'Protected Sites','psSiteL', -1,'Protected
Sites',0,'PS.ProtectedSitesDesig.L', 'Protected Sites - Natura3000 - Desig (Line)',
'ProtectedSite', 'designationscheme = 'natura3000' and designation = 'Desig'', <id2>,
'ps', null ,1, 1);
```

4. Create the hidden sublayer holding geometries of type Multipoint, use a specific definition query, and create a reference to the parent layer created in step 2 (<id2>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer_name, layer_title, spatial_object_type, def_query, parent_id,
spatial_object_type_prefix, layer_keywords, is_hidden, is_visible) VALUES
(<id4>,<id4>,'Protected Sites','psSiteMP', -1,'Protected
Sites',0,'PS.ProtectedSitesDesig.MP', 'Protected Sites - Natura3000 - Desig (Multipoint)',
'ProtectedSite', 'designationscheme = 'natura3000' and designation = 'Desig'', <id2>,
'ps', null, 1, 1);
```

5. Create the hidden sublayer holding geometries of type Point, use a specific definition query, and create a reference to the parent layer created in step 2 (<id2>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer_name, layer_title, spatial_object_type, def_query, parent_id,
spatial_object_type_prefix, layer_keywords, is_hidden, is_visible) VALUES
(<id5>,<id5>,'Protected Sites','psSiteP', -1,'Protected
Sites',0,'PS.ProtectedSitesDesig.P', 'Protected Sites - Natura3000 - Desig (Point)',
'ProtectedSite', 'designationscheme = 'natura3000' and designation = 'Desig'', <id2>,
'ps', null, 1, 1);
```

6. Create the hidden sublayer holding geometries of type Surface, use a specific definition query, and create a reference to the parent layer created in step 2 (<id2>).

```
INSERT INTO layerinfo(objectid, id, theme, fc_name, stype, app_schema, ir_version,
layer name, layer title, spatial object type, def query, parent id,
spatial object type prefix, layer keywords, is_hidden, is_visible) VALUES
(<id6>,<id6>,'Protected Sites','psSites', -1,'Protected
Sites',0,'PS.ProtectedSitesDesig.S', 'Protected Sites - Natura3000 - Desig (Surface)',
'ProtectedSite', 'designationscheme = 'natura3000' and designation = 'Desig'', <id2>,
'ps', null, 1, 1);
```

The SQL statements need some adjustment before they can be used (ID replacement, the specific domain of use, type of database).

2.4 Use Case: Add Scale Dependencies

It is possible to add scale dependencies to the data model. If users want to assign a minimum scale, a maximum scale, and the default visibility to a specific INSPIRE layer or sublayer, they can accomplish that by simply registering the LayerInfo table as versioned (with the option to move edits to the base) and edit this table within ArcMap (see the screen capture below):

- **Layer is default visible in map:** 1 = is visible, 0 = is not visible
- **Minimum scale of this layer:** Enter a numeric value
- **Maximum scale of this layer:** Enter a numeric value

IS_HIDDEN	Layer is default visible in map	Minimum scale of this layer	Maximum scale of this layer
1	1	<Null>	<Null>
1	1	<Null>	<Null>
1	1	<Null>	<Null>
1	1	<Null>	<Null>
1	1	<Null>	<Null>
1	1	<Null>	<Null>
1	1	<Null>	<Null>
0	1	<Null>	<Null>