

Title: Portrayal Extension Supporting Cross-Feature Dependencies

S-100 Maintenance - Change Proposal Form

Organisation	NIWC	Date	2/19/2021
Contact	David Grant	Email	David.Grant1@navy.mil

Change Proposal Type *(Select only one option)*

1.Clarification	2.Correction	3.Extension
		X

Location *(Identify all change proposal locations)*

S-100 Version No.	Part No.	Section No.	Proposal Summary
5.0 draft	9 9a	9-11.2.2 9a-11.2.2.1	Clarify usage of <i>parentId</i> attribute Extend functionality of Parent instruction

Change Proposal

Attached

Change Proposal Justification

Adds a parameter to the 9a-11.2.2.1 **Parent** visibility command. This extension supports the S-101 / ECDIS requirement for expressing cross-feature visibility dependencies that occur when features are associated with one another, such as through a *StructureEquipment* relationship. E.g. A light on a buoy should not be visible when the buoy is not visible.

Cross-feature visibility dependencies can already be expressed using Part 9. The proposed change simply clarifies intended usage of the 9-11.2.2 *parentId* attribute.

What parts of the S-100 Infrastructure will this proposal affect?

- S-100 Feature Concept Dictionary Interface or Database
- S-100 Portrayal Register
- S-100 Feature Catalogue Builder
- S-100 Portrayal Catalogue Builder
- S-100 UML Models
- S-100 GitHub Schemas

Please send completed forms and supporting documentation to the secretary S-100WG.

REDLINES FOR PART 9

9-11.2.2 DrawingInstruction

Role Name	Name	Description	Mult.	Type
Class	DrawingInstruction	Abstract base class for all drawing instructions	-	-
Attribute	id	An identifier for the drawing instruction	0..1	string
Attribute	parentId	Instruction is dependent on a parent drawing instruction(s). <u>If no referenced instructions are executed during rendering then this instruction should not be executed.</u> <u>Execution of referenced instructions can be affected by many aspects of the visualization process including: viewing group settings, display plane visibility, line suppression, scale minimum/maximum, date dependency, hover status, and parent instruction dependencies.</u>	0..1	string
Attribute	hover	Specifies whether the instruction is shown only on hover-over. OEM support for this feature is optional	0..1	boolean
Attribute	viewingGroup	The viewing group the instruction is assigned to	1	string
Attribute	displayPlane	The display plane the instruction is assigned to	1	string
Attribute	drawingPriority	The priority that defines the order of drawing	1	integer
Attribute	scaleMinimum	Scale denominator to define the minimum scale for which the instruction will be shown. If not given there is no minimum scale	0..1	integer
Attribute	scaleMaximum	Scale denominator to define the maximum scale for which the instruction will be shown. If not given there is no maximum scale	0..1	integer
Role	featureReference	The reference to the feature type that will be depicted by the instruction	1	FeatureReference
Role	spatialReference	The reference(s) to the spatial type components of the feature that defines the geometry used for the depiction. Not used when the entire geometry of the feature should be depicted	0..*	SpatialReference
Role	timeValid	The drawing instruction is valid during the specified time interval(s)	0..*	TimeInterval

REDLINES FOR PART 9A

9a-11.2.2.1 Visibility Commands

Visibility commands affect the visibility and drawing order of all subsequent drawing commands. They correspond to attributes of the Part 9 clause 9-11.2.2 *DrawingInstruction* class.

Table 9a-17 – Visibility Commands

Command	Parameters	Type	Initial State	Part 9	Notes
ViewingGroup	viewingGroup	String	""	9-11.1.3	For example: 21000
DisplayPlane	displayPlane	String	""	9-11.1.5	For example: overRadar
DrawingPriority	drawingPriority	Integer	0	9-11.1.6	
ScaleMinimum	scaleMinimum	Integer	max integer	9-11.2.2	
ScaleMaximum	scaleMaximum	Integer	min integer	9-11.2.2	

Id	id	String	""	9-11.2.2	
Parent	id	String	""	9-11.2.25	
	<u>featureID</u>	<u>String</u>	<u>""</u>	<u>9-11.2.2</u>	
Hover	hover	Boolean	false	9-11.2.25	

ViewingGroup: *viewingGroup*

Sets the viewing group for drawing commands which follow.

Applicability: All drawing commands except *NullInstruction*

DisplayPlane: *displayPlane*

Sets the display plane for drawing commands which follow.

Applicability: All drawing commands except *NullInstruction*

DrawingPriority: *drawingPriority*

Sets the drawing priority for drawing commands which follow.

Applicability: All drawing commands except *NullInstruction*

ScaleMinimum: *scaleMinimum*

Sets the scale denominator defining the minimum scale for drawing commands which follow.

Applicability: All drawing commands except *NullInstruction*

ScaleMaximum: *scaleMaximum*

Sets the scale denominator defining the maximum scale for drawing commands which follow.

Applicability: All drawing commands except *NullInstruction*

Id[:*id*]

Sets an identifier for drawing commands which follow. When no parameters are present, resets to the default state of no identifier.

id An identifier for drawing commands which follow

Applicability: All drawing commands

Parent[:*id*,*featureID*]

Visibility of drawing commands which follow is dependent on the visibility of referenced drawing command(s). If no referenced drawing command is executed during rendering then the dependent drawing commands should not be executed.

Execution of referenced drawing commands can be affected by many aspects of the visualization process including: viewing group settings, display plane visibility, line suppression, scale minimum/maximum, date dependency, hover status, and parent drawing command dependencies.

When no parameters are present, resets to the default state of no parent dependency.

id The identifier of the parent drawing command(s)

featureID A unique identifier of a feature instance as described in Part 13. Permits referencing drawing commands that are associated with feature instances other than the current feature instance.

Applicability: All drawing commands except *NullInstruction*

Hover:*hover*

Specifies whether visibility of drawing commands which follow is dependent on hover-over. OEM implementation of support for this feature is optional.

hover

*true*Visibility is dependent on hover-over

false Visibility is not dependent on hover-over

Applicability: All drawing commands except *NullInstruction*