

S-100 – Часть 9а

Изображение (Lua)

Содержание

9a-1 Цель	3
9a-2 Соответствие	3
9a-3 Нормативные ссылки	3
9a-4 Каталог изображений	4
9a-5 Общая модель изображения	4
9a-5.1 Процесс изображения	4
9a-5.2 Процесс изображения Lua	5
9a-5.2.1 Инициализация изображения	6
9a-5.2.2 Генерирование изображения	6
9a-6 Обзор пакета	8
9a-7 Схема ввода данных	8
9a-8 Информационные объекты	8
9a-9 Описательные объекты	8
9a-10 Процесс изображения	8
9a-11 Инструкции по отрисовке	8
9a-11.1 Концепции инструкций по отрисовке	9
9a-11.1.1 Общая концепция	9
9a-11.2 Модель инструкций по отрисовке	9
9a-11.2.1 Команды отрисовки	10
9a-11.2.2 Команды состояния	14
9a-12 Определения условных знаков	38
9a-13 Библиотека изображений	38
9a-14 Специальные функции домена изображений	38
9a-14.1 Функции каталога домена изображений	38
9a-14.1.1 Логический <code>PortrayalMain(String[] featureIDs)</code>	38
9a-14.1.2 <code>void PortrayalInitializeContextParameters(ContextParameter[] contextParameters)</code>	39
9a-14.1.3 <code>ContextParameter PortrayalCreateContextParameter(String contextParameterId, String contextParameterType, String defaultValue)</code>	40
9a-14.1.4 <code>void PortrayalSetContextParameter(String contextParameterId, String value)</code>	40
9a-14.2 Специфические функции домена изображений хоста	41
9a-14.2.1 <code>Boolean HostPortrayalEmit(String featureID, String drawingInstructions, String observedParameters)</code>	41

9а-1 Цель

Эта часть определяет изменения в S-100 Часть 9, необходимые для реализации изображений с использованием механизма сценария, определенного в S-100 Часть 13. Продукты, в которых указывается использование каталога изображений, описанного в этой части, должны также требовать применения S-100 Части 13.

9а-2 Соответствие

Эта часть спецификации соответствует S-100 Часть 13.

9а-3 Нормативные ссылки

Для применения этого документа требуются следующие справочные документы. Для недатированных ссылок применяется последнее издание справочного документа (включая поправки).

Lua 5.1 Reference Manual, <https://www.lua.org/manual/5.1/>

9а-4 Каталог изображений

В описание каталога изображений Части 9 изменений нет.

9а-5 Общая модель изображения

В общую модель изображения части 9 изменений нет. Изображение Lua следует общей модели изображения, описанной в пункте 9-5. Рисунок 9а-1 показывает общую модель изображения.

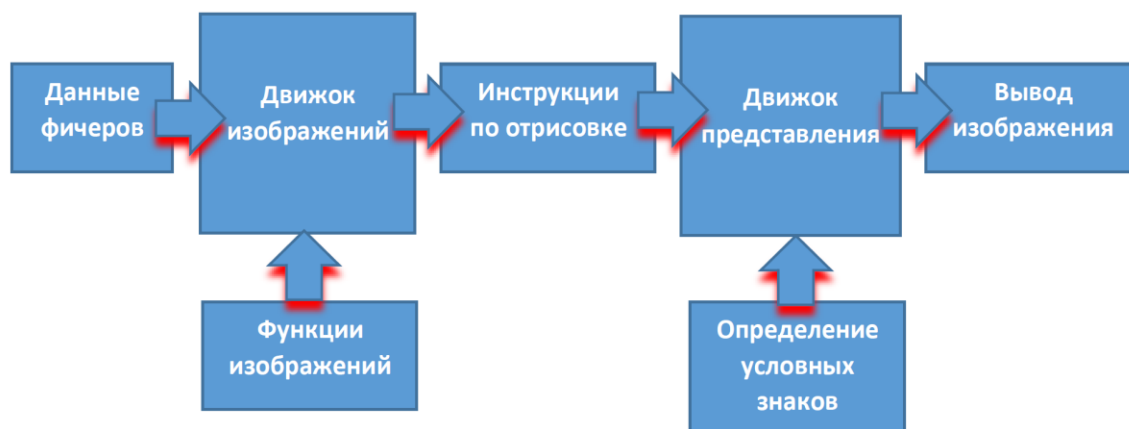


Рисунок 9а-1 – Общая модель изображения

9а-5.1 Процесс изображения

Как показано на рисунке 9а-2, изображение Lua требует внесения следующих изменений в процесс изображения, описанный в пункте 9-5.1 части 9 и отраженный в таблице 9а-1:

Таблица 9а-1 – Изменения в процесс изображения

Часть 9	Часть 9а
Функции изображения написаны на языке программирования XSLT	Функции изображения написаны на языке программирования Lua
Host предоставляет реализацию XSLT.	Host предоставляет интерпретатор Lua или виртуальную машину Lua.
Данные фичеров обрабатываются функциями изображения через XML-документ, который должен описывать все фичеры, подлежащие изображению, вместе со всей атрибутикой, пространственными отношениями, ассоциациями информации и всей другой информации, которая может быть использована для изображения фичеров.	Данные фичеров изначально не обрабатываются функциями изображения. Вместо этого хост предоставляет список идентификаторов фичеров для отображения; функции изображения будут требовать атрибутику, пространственные отношения, ассоциации информации и всю другую информацию по мере необходимости через обратный вызов функций хоста.
Инструкции по отрисовке возвращаются хосту как документ XML, являющийся результатом преобразования XSL применяемого для входных данных фичера.	Инструкции по отрисовке возвращаются хосту через функцию хоста отзыва <i>HostPortrayalEmit</i> .

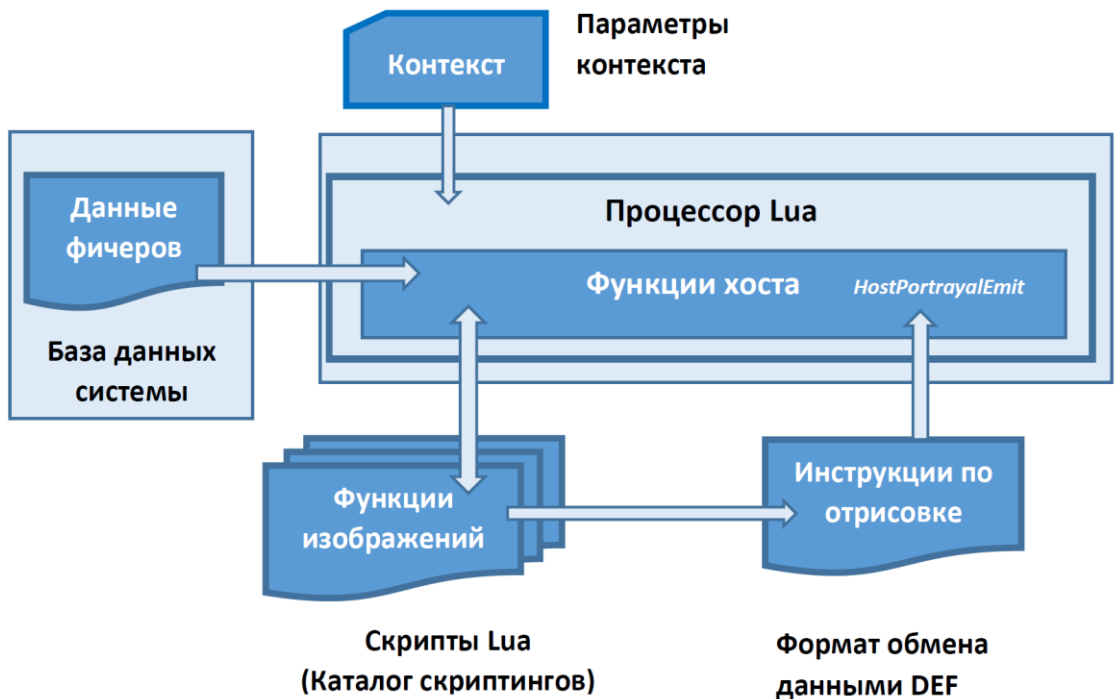


Рисунок 9а-2 – Процесс изображения

9а-5.2 Процесс изображения Lua

В этом разделе подробно описывается процесс изображения части 9а и указывается, где есть изменения в части 9. Процесс изображения Lua показан на рисунке 9а-3.



Рисунок 9а-3 – Процесс изображения Lua

9а-5.2.1 Инициализация изображения

Перед вызовом функций изображения Lua хост должен зарегистрировать функции доменного каталога сценариев, загрузив файл правил каталога изображений *TopLevelTemplate* (файл скриптов Lua). Чтобы предотвратить коллизии имен на *PortrayalMain*, хост должен создавать экземпляры и инициализировать новую среду исполнения Lua каждый раз, когда *TopLevelTemplate* изменяется. В качестве альтернативы, хост может поддерживать несколько запусков Lua, по одному для каждого *TopLevelTemplate*.

После регистрации функций каталога сценариев хост вызывает *PortrayalInitializeContextParameters*, передавая имя и значение по умолчанию для каждого параметра контекста изображения, определяемого каталогом изображений. Значения параметров контекста связаны с данным набором данных и остаются в силе до закрытия сессии сценариев или изменения значений через *PortrayalSetContextParameter*.

9а-5.2.2 Генерирование изображения

Функция сценария изображения *PortrayalMain* (см. раздел 9а-14.1.1) используется для создания инструкций по отрисовке набора реализаций фичеров. Хост передает набор идентификаторов фичеров на *PortrayalMain*; сценарии отображения будут перебирать идентификаторы фичеров и генерировать инструкции по отрисовке каждого.

По мере обработки каждой реализации фичера механизм отображения будет вызывать стандартные функции хоста для запроса атрибута, пространственной или другой информации по мере необходимости. После завершения обработки реализации фичера механизм отображения вызовет *HostPortrayalEmit* (см. пункт 9а-14.2.1) и предоставит инструкции по отрисовке этой реализации фичера приложения хоста.

Отображение данного *S100_Dataset* завершается, когда возвращается вызов *PortrayalMain*. Если изображение успешно завершено, *PortrayalMain* возвращает true, в противном случае *PortrayalMain* возвращает false вместе с сообщением, указывающим, почему изображение не было запущено до завершения.

Хост может прекратить представление перед обработкой всех реализаций фичеров, вернув false из *HostPortrayalEmit*.

Вызов *PortrayalMain* со всеми ID фичеров из данного набора данных создаст инструкции по отрисовке для всего набора данных. Инструкции по отрисовке подмножества набора данных могут (ре)генерироваться путём передачи идентификаторов фичеров, соответствующих подмножеству. Это полезно, когда хосту нужно регенерировать набор кэшированных инструкций отрисовки, или если хост изображает подмножество набора данных, например, один *S100_DataCoverage*.

9а-5.2.2.1 Применение кэша изображений

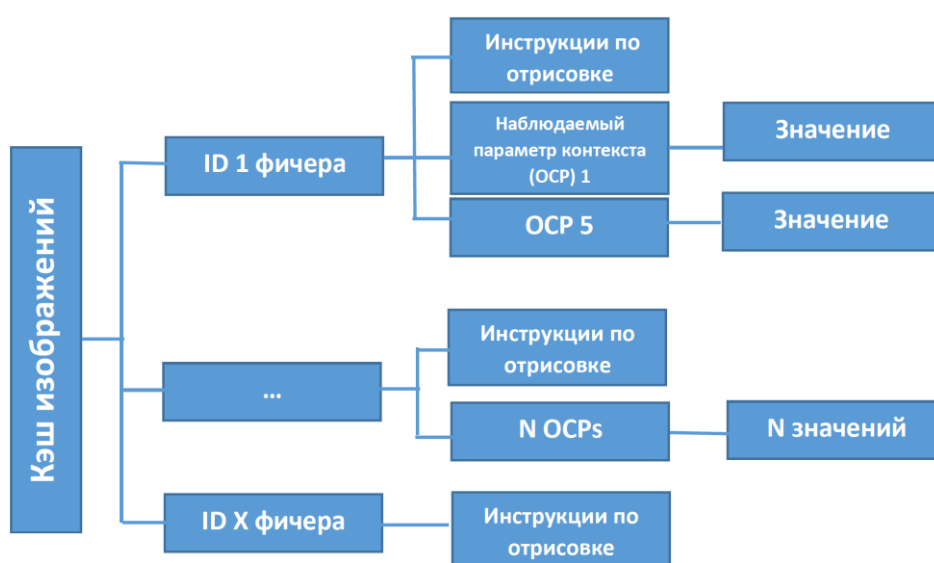
Для ускорения процесса рендеринга хост может дополнительно реализовывать кэш изображения. Кэш изображения используется для кэширования инструкций по отрисовке, которые выводятся из изображения.

Кэширование инструкций по отрисовке для каждой реализации фичера позволяет хосту повторно отображать реализации фичера без повторного создания их изображения. Кэшированная инструкция по отрисовке должна регенерироваться только в том случае, если один или несколько параметров контекста, которые использовались для генерации инструкции по отрисовке, изменяются.

Когда сценарии изображения возвращают инструкции по отрисовке для реализации фичеров, они также возвращают список "наблюдаемых" параметров контекста изображения (см. пункт 9а-14.2.1). Наблюдаемые параметры контекста - это те параметры контекста, которые были оценены во время генерации инструкций по отрисовке для конкретного фичера. Более подробная информация о параметрах контекста содержится в пункте 9-13.3.22 Части 9.

Условный кэш изображения показан на рисунке 9а-4. Для использования, хост должен кэшировать значение наблюдаемых параметров контекста вместе с генерируемыми инструкциями по отрисовке и ассоциировать их с реализацией фичера. Обратите внимание, что реализация фичера может иметь любое количество наблюдаемых контекстных параметров, включая ноль.

Любые изменения в параметре контекста требуют, чтобы хост регенерировал инструкции по отрисовке для всех реализаций фичеров с соответствующим наблюдаемым параметром контекста. В качестве альтернативы хост может использовать кэшированные инструкции по отрисовке, которые ранее были сгенерированы для нового значения изменённого параметра(ов) контекста. Фичеры, не имеющие наблюдаемых параметров, могут сохраняться в кэше до выпуска нового каталога изображений.



ОСР – Observed Context Parameter (Наблюдаемый параметр контекста)

Рисунок 9а-4 - Условное изображение кэша

9a-5.2.2.2 Предварительная обработка изображения

Применение кэша изображения позволяет хосту предварительно генерировать инструкции по отрисовке для заданного набора или наборов контекстных параметров. Это, как правило, будет использовано как часть функциональности импорта данных хостов.

9a-6 Обзор пакета

Обзор пакета Части 9 не изменился, хотя большинство пакетов в Части 9a не используются из-за удаления схемы ввода изображения.

9a-7 Схема ввода данных

Эта часть не использует схему ввода данных, как определено в пункте 9-7 части 9. Данные передаются между изображением 9a и хостом, как описано в части 13.

9a-8 Информационные объекты

Информационные объекты, описанные в части 9, не используются в части 9a. Вместо этого информация, связанная с фичерами, которые подлежат изображению, получена в соответствии с описанием, содержащимся в части 13.

9a-9 Описательные объекты

Объекты фичеров, описанные в Части 9, не используются в Части 9a. Вместо этого все фичеры извлекаются из хоста, как описано в Части 13.

9a-10 Процесс изображения

Обработка XSLT, описанная в пункте 9-10 части 9, заменена на Lua, как описано в части 13.

9a-11 Инструкции по отрисовке

Инструкции по отрисовке предоставляются хосту с использованием DEF, как описано в части 13 пункт 13.6.1. Одна инструкция по отрисовке эквивалентна одному элементу DEF.

В этом разделе описаны модель и схема инструкций по отрисовке.

9a-11.1 Концепции инструкций по отрисовке

9a-11.1.1 Общая концепция

Как и в части 9, выход движка изображений представляет собой набор инструкций по отрисовке. Они обычно связывают реализацию фичера со ссылкой на условный знак. Геометрия либо взята из типа фичера, либо может быть сгенерирована функциями изображений. Последние подкрепляются концепцией расширенной геометрии, описанной в пункте 9-11.1.12 Части 9.

Концептуальная модель для инструкций по отрисовке части 9a представляет собой управляемый по команде механизм состояния. Эта модель согласуется как с SVG, так и с S-52 DAI, но отличается от части 9, в которой используются инструкции по отрисовке без состояния.

Для использования инструкций по отрисовке части 9a хост должен поддерживать состояние при выполнении инструкций по отрисовке для данной реализации фичера. Например, если инструкция по отрисовке устанавливает цвет ручки, этот цвет ручки также должен использоваться для последующих инструкций по отрисовке. Перед выполнением инструкций по отрисовке для каждой реализации фичера состояние должно быть перезагружено.

9a-11.2 Модель инструкций по отрисовке

Как и в части 9, в настоящем разделе описываются выходные данные функций изображения. Одна специфическая функция домена хоста сценариев (смотрите в пункте 9a-14.2.1 *HostPortrayalEmit*) предоставляет инструкции по отрисовке для каждой реализации фичера.

Каждая инструкция по отрисовке закодирована в элементе DEF, как описано в пункте 13-6.1 Части 13. Инструкция по отрисовке — это упорядоченная пара, состоящая из команды и списка параметров. Команда кодируется в элементе DEF, а параметры команд кодируются в списке параметров DEF.

Таблица 9a-2 – DEF кодирование инструкций по отрисовке

Элемент изображения	DEF кодирование	Пример
Инструкция по отрисовке	Элемент	FillColor:CHBRN,0
Команда	Пункт	FillColor
Список параметров	Список параметров	CHBRN,0
Параметр	Параметр	CHBRN

Каждая инструкция по отрисовке содержит одну команду, зависящую от регистра. Каждая команда имеет ноль или более параметров.

Существует два типа команд: команды отрисовки и команды состояния. Команды отрисовки предписывают хосту отображать графику. Команды состояния поручают хосту устанавливать состояние для последующих команд отрисовки.

Каждая команда и ее параметры описываются в следующих подразделах, сгруппированных по целям. В нижеследующих таблицах столбец Тип описан в таблице 13-7 части 13. Колонка **XRef** относится к эквивалентной концепции

инструкции по отрисовке части 9. Ссылка на часть 9 может содержать соответствующую информацию, такую как диапазон ожидаемых значений или единицы измерения.

9а-11.2.1 Команды отрисовки

Команды отрисовки используются для отображения графики. Они аналогичны реализации пункта 9-11.2 части 9 класса *DrawingInstruction*. Команды отрисовки перечислены в таблице 9а-3, и каждая команда описана на следующих страницах.

Таблица 9а-3 – Команды отрисовки

Команда	Параметры	Тип параметра	Пункт части 9
PointInstruction	symbol	string	9-11.2.7 9-11.2.13
LineInstruction	lineStyle	string	9-11.2.8 9-11.2.15 9-11.2.16
LineInstructionUnsuppressed	lineStyle	string	9-11.2.8 9-11.2.15 9-11.2.16
ColorFill	Token	string	9-11.2.9
	transparency	double	9-12.5.1.4 9-11.2.17
AreaFillReference	reference	string	9-11.2.9 9-12.5.1.3 9-11.2.17
PixmapFill	reference	string	9-11.2.9 9-12.5.1.5 9-11.2.17
SymbolFill	symbol	string	9.11.2.9
	v1	Vector	9-12.5.1.6
	v2	Vector	9-11.2.17
	clipSymbols	boolean	
HatchFill	direction	Vector	9.11.2.9
	distance	double	9-12.5.1.7
	lineStyle	string	9-11.2.17
TextInstruction	text	string	9-11.2.10 9-11.2.12
CoverageFill	attributeCode	string	9-11.1.12
	uom	string	9-11.2.11
	placement	string	9-12.7.4.7
NullInstruction	-	-	9-11.2.6

Графическое отображение каждой команды отрисовки может быть изменено предыдущими командами состояний, как описано в пункте 9а-11.2.2.

PointInstruction: *symbol*

Поручает хосту нарисовать условный знак из Каталога Изображений, размещенный следующим образом:

Таблица 9а-4 – Размещение условного знака PointInstruction

Тип геометрии	Размещение символа
Point	В точке, затем применяется <i>LocalOffset</i>
Line	Вдоль линии с помощью <i>LinePlacement</i> , затем применяется <i>LocalOffset</i>
Area	В <i>AreaCRS</i> , затем применяется <i>LocalOffset</i> . Обратите внимание, что это может привести к тому, что символ будет нарисован в нескольких местах

LineInstruction: *lineStyle[,lineStyle,...]*

Поручает хосту построить геометрию линии или области с использованием заданного *linestyle(s)*. Хост должен обеспечить, чтобы линейные сегменты с меньшим приоритетом рисования были исключены (не нарисованы), когда они совпадают с отрезками линий с более высоким приоритетом отрисовки.

Каждый параметр линии относится либо к стилю, определенному в Каталоге изображений, либо к стилю линии, созданному предыдущей командой *LineStyle*.

Примечание: Пункт 10а-5.10.1 части 10 определяет, как скрытые пространственные элементы кодируются в наборе данных. При выполнении этой инструкции хост должен подавлять изображение скрытых пространственных элементов.

LineInstructionUnsuppressed: *lineStyle[,lineStyle,...]*

Поручает хосту построить геометрию линии или области с использованием заданного *linestyle(s)*.

Линейные сегменты должны быть нарисованы без учета совпадающих линейных сегментов.

Каждый параметр *linestyle* относится либо к стилю, определенному в Каталоге изображений, либо к стилю линии, созданному предыдущей командой *LineStyle*.

Примечание: Пункт 10а-5.10.1 части 10 определяет, как скрытые пространственные элементы кодируются в наборе данных. При выполнении этой инструкции хост должен подавлять изображение скрытых пространственных элементов.

ColorFill:token[,transparency]

Поручает хосту заполнить область с помощью данного маркера цвета и прозрачности. Если прозрачность не задана, предполагается, что значение ноль.

AreaFillReference:reference

Поручает хосту заполнить область с помощью areaFill (пункт 9-13.3.9 части 9), определенного в Каталоге изображений.

PixmapFill:reference

Поручает хосту заполнить область с помощью pixmap (пункт 9-13.3.5 части 9), определенного в Каталоге изображений.

Предыдущая команда AreaCRS может установить начальные параметры шаблона.

SymbolFill:symbol,v1,v2

Поручает хосту заполнить область с помощью условного знака, определенного в Каталоге изображений. Предыдущая команда AreaCRS может установить начальные параметры шаблона.

symbol Условный знак, используемый шаблоном.

v1 Смещение следующего условного знака относительно первого шаблона в соответствии с местной CRS.

v2 Смещение следующего условного знака относительно второго шаблона в соответствии с местной CRS.

clipSymbols Если true (или отсутствует), условные знаки обрезаются на границах области. Если false, условные знаки, выходящие за границы района, вообще не рисуются.

HatchFill:direction,distance,lineStyle[,lineStyle]

Поручает хосту заполнить область с помощью нового условного знака, определенного в Каталоге изображений. Направление и расстояние определяются как в пункте 9-12.5.1.8 части 9.

Каждый параметр linestyle относится либо к стилю, определенному в Каталоге изображений, либо к стилю линии, созданному предыдущей командой *LineStyle*.

Предыдущая команда *AreaCRS* может установить исходные параметры шаблона.

direction Вектор, определяющий направление набора прямых.

distance Расстояние между линиями, измеряемое перпендикулярно направлению.

lineStyle Ссылка на стиль линии, используемый для каждой новой линии.

TextInstruction:text

Поручает хосту нарисовать указанный текст, размещенный следующим образом:

Таблица 9а-5 – Начальное размещение TextInstruction

Тип геометрии	Начальное размещение
Point	Относительно точки
Line	Относительно линии, как определено <i>LinePlacement</i>
Area	Относительно <i>AreaCRS</i> . Обратите внимание, что это может привести к тому, что текст будет нарисован в нескольких местах

После определения исходного положения текст смещается в соответствии с командами состояния *LocalOffset* и *TextVerticalOffset*. Текст выравнивается в соответствии с командами состояния *TextAlignHorizontal* и *TextAlignVertical*.

Если ему предшествует команда *FontReference*, шрифт указывается в каталоге изображений. В противном случае хост должен построить шрифт, используя значения, заданные предыдущими командами состояния *FontColor*, *FontSize*, *FontPropart*, *FontWeight*, *FontSlant*, *FontSerifs* и *FontStrikethro*.

text Текст для отображения.

CoverageFill:attributeCode[,uom[,placement]]

Поручает хосту заполнить покрытие с помощью элементов таблицы просмотра, созданных с помощью команды состояния *LookupEntry*. Хост должен очистить список поиска покрытия после завершения.

attributeCode Указывает атрибуты фичеров, используемые таблицей просмотра.

uom Если имеется, указывает единицы измерения для диапазона значений в таблице просмотра. Если он отсутствует, то значения диапазона и значения атрибутов имеют те же единицы измерения, как определено в каталоге фичеров.

placement Если имеется, указывает размещение знака/текстовых аннотаций

ПРИМЕЧАНИЕ: При привязке оповещений к значениям покрытия могут присутствовать или не присутствовать элементы описания в списке просмотра покрытия.

NullInstruction

Используется для указания что фичер не изображается намеренно. Может ассоциировать оповещение с фичером или геометрией, которая не отображается, или геометрией оповещения, отличающегося от изображаемой геометрии.

9a-11.2.2 Команды состояния

Команды состояния используются для установки или изменения состояния для рисования следующих за ними команд. Для реализации изображения хост должен связать каждый параметр команды состояния с переменной; каждая команда состояния изменяет значение одной или нескольких из этих переменных.

Хост должен установить начальное состояние, как указано в таблицах следующих подразделов. Состояние должно быть сброшено перед выполнением инструкций по отрисовке для каждой реализации фичера.

Для каждой команды состояния, перечисленной в следующих подразделах, указывается применимость; это обозначает какие команды используют переменные, заданные командой состояния.

Таблица 9a-6 показывает различные типы команд состояния.

Таблица 9a-6 – Типы команд состояния

Тип команды	Команда	Назначение
Visibility	ViewingGroup	Изменяет видимость и порядок рисования команды отрисовки
	DisplayPlane	
	DrawingPriority	
	ScaleMinimum	
	ScaleMaximum	
	Id Parent Hover	
Transform	LocalOffset	Применяет преобразования к элементам, нарисованным с помощью команд отрисовки
	LinePlacement	
	AreaPlacement	
	AreaCRS	
	Rotation	
	ScaleFactor	
Line Style	LineStyle	Определяет linestyles для использования при рисовании команд
	LineSymbol	
	Dash	
Text Style	FontColor	Изменяет внешний вид текста, нарисованного с помощью команд отрисовки
	FontSize	
	FontProportion	
	FontWeight	
	FontSlant	
	FontSerifs	
	FontUnderline	
	FontStrikethrough	
	FontUpperline	
	FontReference	
	TextAlignHorizontal	

	TextAlignVertical	
	TextVerticalOffset	
Colour Override	OverrideColor	Заменяет цвета, определенные внутри условного знака или рixmap, на который ссылаются команды отрисовки
	OverrideAll	
Geometry	SpatialReference	Определяет новую геометрию (дополненная геометрия) или ограничивает геометрию, используемую командой отрисовки
	AugmentedPoint	
	AugmentedRay	
	AugmentedPath	
	Polyline	
	Arc3Points	
	ArcByRadius	
	Annulus	
ClearGeometry		
Coverage	LookupEntry	Определяет элементы таблицы просмотра, на которые может ссылаться команда отрисовки <i>CoverageFill</i>
	NumericAnnotation	
	SymbolAnnotation	
	CoverageColor	
Time	Date	Определяет интервалы времени
	Time	
	DateTime	
	TimeValid	
	ClearTime	
Alert	AlertReference	Ассоциирует геометрию с каталогом оповещений

9a-11.2.2.1 Команды видимости

Команды видимости влияют на видимость и порядок рисования всех последующих команд отрисовки. Они соответствуют атрибутам раздела 9-11.2.2 части 9 *DrawingInstruction* class.

Таблица 9a-7 – Команды видимости

Команда	Параметры	Тип	Начальное состояние	Часть 9	Примечания
ViewingGroup	viewingGroup	string[]	""	9-11.1.3	For example: 21000
	...				
	viewingGroupN				
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	""	9-11.2.2	
Hover	hover	boolean	false	9-11.2.2	

ViewingGroup: *viewingGroup[,viewingGroup2[,...]]*

Задает группу(ы) просмотра для команд отрисовки. Команда отрисовки отключена, если группа просмотра отключена.

Применение: Все команды отрисовки

DisplayPlane: *displayPlane*

Задает плоскость отображения для рисования последующих команд.

Применение: Все команды отрисовки кроме *NullInstruction*

DrawingPriority: *drawingPriority*

Устанавливает приоритет рисования для последующих команд отрисовки.

Применение: Все команды отрисовки кроме *NullInstruction*

ScaleMinimum: *scaleMinimum*

Задает знаменатель масштаба, определяющий минимальный масштаб для последующих команд отрисовки.

Применение: Все команды отрисовки кроме *NullInstruction*

ScaleMaximum: *scaleMaximum*

Задает знаменатель масштаба, определяющий максимальный масштаб для следующих команд отрисовки.

Применение: Все команды отрисовки кроме *NullInstruction*

Id[:*id*]

Устанавливает идентификатор для следующих команд. При отсутствии параметров, восстанавливает состояние по умолчанию без идентификатора.

id Идентификатор следующей команды отрисовки

Применение: Все команды отрисовки

Parent[:*id*]

Видимость следующих команд отрисовки зависит от указанной команды (команд) отрисовки. Если во время рендеринга указанная команда отрисовки не выполняется, то зависимые команды отрисовки не должны выполняться.

Для выражения кросс-фичерных зависимостей, указанная команда (команды) может быть связана с другой реализацией фичера, отличной от текущей реализации фичера; изучайте все команды отрисовки для всех реализаций фичеров при определении родительской команды (команд) отрисовки.

Выполнение ссылочных (родительских) команд отрисовки может быть затронуто многими аспектами процесса визуализации, включая настройки группы просмотра, видимости плоскости отображения, подавления линии, минимального/максимального масштаба, зависимости от даты, состояния курсора и зависимости от родительской команды отрисовки.

При отсутствии параметров, восстанавливается в состояние по умолчанию без родительской зависимости.

id Идентификатор родительской команды (команд) отрисовки

Применение: Все команды отрисовки кроме *NullInstruction*

Hover:*hover*

Указывает, зависит ли видимость последующих команд рисования от наведения курсора. Реализация поддержки этой функции является необязательной.

hover

true Видимость зависит от наведения курсора

false Видимость не зависит от наведения курсора

Применение: Все команды отрисовки кроме *NullInstruction*

9a-11.2.2.2 Команды трансформации

Преобразуют команды, применяя трансформацию элементов, таких как условные знаки, полученные с помощью последующих команд отрисовки.

Таблица 9а-8 – Команды трансформации

Команда	Параметры	Тип	Начальное состояние	Пункт Части 9
LocalOffset	xOffsetMM	double	0	9-12.2.2.7
	yOffsetMM	double	0	
LinePlacement	linePlacementMode	string	Relative	9-12.3.1.5
	offset	double	0.5	
	endOffset	double	nil	
AreaPlacement	areaPlacementMode	string	VisibleParts	9-12.3.1.6
AreaCRS	areaCRSType	string	GlobalGeometry	9-12.5.1.9
Rotation	rotationCRS	string	PortrayalCRS	9-12.2.2.7
	rotation	double	0	9-12.3.1.1 9-12.4.1.4 9-12.6.3.5
ScaleFactor	scaleFactor	double	1.0	9-12.2.2.7 9-12.3.1.1 9-12.4.1.4

LocalOffset: *xOffsetMM, yOffsetMM*

Указывает смещение от географического положения с помощью локальных CRS, применяемых к последующим командам отрисовки.

Применение: *PointInstruction, SymbolFill, TextInstruction*

LinePlacement: *linePlacementMode, offset[, endOffset]*

Указывает размещение в строке условных знаков или вывода текста с помощью последующих команд отрисовки.

linePlacementMode

Relative *offset* в относительных координатах, 0 для начала и 1 для конца кривой.

Absolute *offset*, определяющий расстояние от начала кривой.

offset Смещение от начала кривой. Если *endOffset* также указан, это определяет начальную точку команды *TextInstruction*.

endOffset Если присутствует, указывает, что текст, нарисованный любым *TextInstruction*, располагается вдоль линейной геометрии. Если присутствует и содержит значение, далее указывает конечную точку любой команды *TextInstruction*. Горизонтальное выравнивание в данном случае эффекта не имеет.

Применение: *PointInstruction, TextInstruction*

AreaPlacement:*areaPlacementMode*

Указывает положение в области площадного условного знака или вывод текста с помощью последующих команд отрисовки.

areaPlacementMode – один из:

VisibleParts Условный знак или текст должны быть размещены в репрезентативном положении в каждой видимой части поверхности.

Geographic Условный знак или текст помещаются в репрезентативное положение географического объекта.

Применение: *PointInstruction, TextInstruction*

AreaCRS:*areaCRSType*

Указывает способ привязки выходных шаблонов заливки с помощью последующих команд отрисовки.

areaCRSType – один из:

Global Точка привязки соответствует расположению на устройстве рисования, например, начиная с угла экрана. По мере того, как движется экран, шаблон будет отображаться, перемещаясь/двигаясь на экране.

LocalGeometry Точка привязки соответствует локальной геометрии изображаемого объекта, например, верхнему левому углу объекта. Шаблоны соседних объектов могут не совпадать.

GlobalGeometry Точка привязки шаблона заливки определяется в общем месте, так что шаблоны остаются последовательными относительно всех площадных объектов.

Применение: *AreaFillReference, PixmapFill, SymbolFill, HatchFill, TextInstruction*

Rotation:*rotationCRS,rotation*

Указывает угол поворота условных знаков или вывода текста с помощью последующих команд отрисовки.

rotationCRS – один из:

GeographicCRS Географическая CRS с широтой и долготой, измеренными в градусах. Вращение определяется по часовой стрелке от направления истинного севера.

<i>PortrayalCRS</i>	Декартова система координат с осью У, направленной вверх. Вращение определяется в градусах по часовой стрелке от положительной оси У.
<i>LocalCRS</i>	Декартова система координат, построенная на локальной геометрии. Вращение в градусах по часовой стрелке от положительной оси у.
<i>LineCRS</i>	Не-декартова система координат, в которой ось х соответствует геометрии кривой, а ось у перпендикулярна оси х (положительная слева от оси х). Единицы измерения по осям и для расстояний - миллиметры. Углы измеряются в градусах по часовой стрелке от положительной оси У.

Смотрите подробности в Части 9 раздел 9-12.2.2.7.

Применение: *PointInstruction*, *SymbolFill*, *TextInstruction*, *CoverageFill*

ScaleFactor: *scaleFactor*

Указывает масштабный коэффициент, применяемый к условным знакам или выводимому тексту последующими командами отрисовки.

Применение: *PointInstruction*, *SymbolFill*, *TextInstruction*, *CoverageFill*

9а-11.2.2.3 Команды стилей линии

Команды стилей линии создают стили, на которые могут ссылаться последующие команды отрисовки. Эти команды являются частью функциональности пакета *LineStyle*, описанного в пункте 9-12.4 части 9.

Таблица 9а-9 – Команды *LineStyle*

Команда	Параметры	Тип	Начальное состояние	Пункт Части 9	Примечания
Dash	start	double	-	9-12.4.1.3	Units: millimetres
	length	double	-		
LineStyleSymbol	reference	string	-	9-12.4.1.4	
	position	double	-		
	rotation	double	0		
	crsType	CRSType	LocalCRS		
	scaleFactor	double	1.0		
LineStyle	name	string	-	9-12.4.1.1	
	intervalLength	double	-	9-12.4.1.2	
	width	double	-		
	token	string	-		
	transparency	double	0		
	capStyle	string	Butt		
	joinStyle	string	Bevel		
	offset	double	0.0		

Dash: *start, length*

Задаёт шаблон штриха для одной последующей команды *LineStyle*. Может быть повторена, чтобы указать, что несколько шаблонов штриха применяются к одной команде *LineStyle*.

ПРИМЕЧАНИЕ: Эта команда не устанавливает состояние для любой команды отрисовки; она только устанавливает состояние для команды *LineStyle*.

start Начало штриха, измеренное от начала линии вдоль оси x линейного CRS (единицы в миллиметрах).

length Длина штриха вдоль оси x линейной CRS (единицы в миллиметрах).

Applicability: *LineStyle*

LineStyleSymbol: *reference, position[, rotation[, crsType[, scaleFactor]]]*

Определяет использование условного знака для одной последующей команды *LineStyle*. Может быть повторена, чтобы указать, что несколько условных знаков применяются к команде *LineStyle*.

reference Ссылка на внешнее определение графического изображения. Это относится к идентификатору элемента каталога изображений.

position Положение условного знака, измеренное с начала повторяющегося интервала вдоль оси x линейной CRS (единицы в миллиметрах).

rotation Угол поворота условного знака.

crsType Тип CRS, в которую должен быть преобразован условный знак. Возможны значения *LocalCRS* и *LineCRS*.

scaleFactor Масштабный множитель условного знака.

Applicability: *LineStyle*

LineStyle:name,intervalLength,width,token[,transparency[,capStyle[,joinStyle[,offset]]]]

Создает именованный стиль линии для использования в последующих командах отрисовки. Может предшествовать нулем или другими командами *Dash* и/или *LineStyle*, которые применяются к линии. Если команды *Dash* не предшествуют команде *LineStyle*, то создается сплошная линия.

name Имя, назначенное для линии и используемое для ссылки на стиль линии из *LineInstruction*. В случае противоречия имени стиля линии из Каталога изображений и командой *LineStyle*, команда *LineStyle* имеет приоритет.

intervalLength Длина повторяющегося интервала стиля линии вдоль оси *x* линейной CRS (единицы в мм). Может быть пропущена, если определена сплошная линия.

width Ширина пера в мм, используемого для рисования этого стиля линии.

token Указывает цвет, используемый для рисования этого стиля линии.

transparency Указывает прозрачность, используемую для рисования этого стиля линии.

capStyle Обозначение, которое применяется, когда заканчивается участок линии. Может быть *Butt*, *Square* или *Round*. См. часть 9 раздел 9-12.4.1.8 *CapStyle*.

joinStyle Обозначение, которое применяется, когда два линейных сегмента стыкуются. Может быть *Bevel*, *Miter* или *Round*. См. часть 9 раздел 9-12.4.1.7 *JoinStyle*.

offset Смещение, перпендикулярное направлению линии. Значение относительно оси *y* линейной CRS (положительный слева, миллиметры).

Applicability: *LineInstruction*, *LineInstructionUnsuppressed*, *HatchFill*

9а-11.2.2.4 Команды стиля текста

Команды стиля текста изменяют внешний вид текста, изображаемого последующими командами рисования.

Таблица 9а-10 – Команды стиля текста

Команда	Параметры	Тип	Начальное состояние	Часть 9	Примечания
Command	Parameters	Type	Initial State	Part 9	Notes
FontColor	token	String	""	9-12.6.3.8	Opaque
	transparency	Double	0	9-12.2.2.3	
FontBackgroundColor	token	String	""	9-12.6.3.8	Transparent
	transparency	Double	1	9-12.2.2.3	
FontSize	bodySize	Double	10	9-12.6.3.8	
FontProportion	proportion	String	Proportional	9-12.6.3.11	
FontWeight	weight	String	Medium	9-12.6.3.10	
FontSlant	slant	String	Upright	9-12.6.3.9	
FontSerifs	serifs	Boolean	false	9-12.6.3.2	
FontUnderline	underline	Boolean	false	9-12.6.3.12	
FontStrikethrough	strikethrough	Boolean	false	9-12.6.3.12	
FontUpperline	upperline	Boolean	false	9-12.6.3.12	
FontReference	fontReference	String	""	9-12.6.3.3	
TextAlignHorizontal	horizontalAlignment	String	Start	9-12.6.3.14	
TextAlignVertical	verticalAlignment	String	Baseline	9-12.6.3.13	
TextVerticalOffset	verticalOffset	Double	0	9-12.6.3.8	

FontColor: *token*[,*transparency*]

Указывает цвет и прозрачность для глифов, нарисованных последующими командами отрисовки.

Applicability: *TextInstruction*

FontBackgroundColor: *token*[,*transparency*]

Указывает цвет и прозрачность, используемые для заливки прямоугольника, окружающего текст, нарисованный последующими командами отрисовки.

Applicability: *TextInstruction, CoverageFill*

FontSize: *bodySize*

Указывает размер в точках для текста, нарисованного последующими командами отрисовки.

Applicability: *TextInstruction, CoverageFill*

FontProportion: *proportion*

Указывает пропорцию шрифта, используемую для текста, нарисованного последующими командами отрисовки.

proportion – одна из:

MonoSpaced Фонт, в котором все типы шрифтов имеют одинаковую ширину, должен быть выбран. Также известны как печатные шрифты.

Proportional Следует выбрать фонт, в котором каждый тип шрифта может иметь разную ширину.

Applicability: *TextInstruction, CoverageFill*

FontWeight: *weight*

Указывает толщину шрифта для текста, нарисованного последующими командами отрисовки.

weight – одна из:

Light Тип шрифта тонкий (стандартный).

Medium Тип шрифта толще, чем *Light*, но не толще, чем *Bold*.

Bold Тип шрифта жирный (**Bold**).

Applicability: *TextInstruction, CoverageFill*

FontSlant: *slant*

Указывает наклон, используемый для текста, нарисованного последующими командами отрисовки.

slant – один из:

Upright Тип шрифта вертикальный.

Italics Тип шрифта курсив с наклоном вправо.

Applicability: *TextInstruction, CoverageFill*

FontSerifs: *serifs*

Указывает, должен ли шрифт, используемый для текста, нарисованного последующими командами отрисовки, содержать насечки.

Applicability: *TextInstruction, CoverageFill*

FontUnderline:*underline*

Указывает, следует ли подчеркивать текст, нарисованный последующими командами отрисовки.

Applicability: *TextInstruction*

FontStrikethrough:*strikethrough*

Указывает, должен ли текст, нарисованный последующими командами отрисовки, отображаться с перечеркнутой линией по центру текста.

Applicability: *TextInstruction*

FontUpperline:*upperline*

Указывает, должен ли текст, нарисованный последующими командами отрисовки, отображаться строкой над текстом.

Applicability: *TextInstruction*

FontReference:*fontReference*

Указывает что текст, нарисованный последующими командами отрисовки, должен быть изображен с использованием указанного шрифта из Каталога Изображений. *fontReference* - идентификатор внешнего файла в каталоге изображений.

Applicability: *TextInstruction*

TextAlignHorizontal:*horizontalAlignment*

Указывает размещение текста относительно точки привязки в горизонтальном направлении для последующих команд отрисовки.

horizontalAlignment – один из:

<i>Start</i>	Точка привязки находится в начале текста.
<i>Center</i>	Точка привязки находится в (горизонтальном) центре текста.
<i>End</i>	Точка привязки находится в конце текста.

Applicability: *TextInstruction*

TextAlignVertical:*verticalAlignment*

Указывает размещение текста относительно точки привязки в вертикальном направлении для последующих команд отрисовки.

verticalAlignment – один из:

- Top* Точка привязки находится в верхней части квадрата.
- Center* Точка привязки находится в (вертикальном) центре квадрата.
- Bottom* Точка привязки находится в нижней части квадрата.

Applicability: *TextInstruction*

TextVerticalOffset: *verticalOffset*

Задаёт вертикальное смещение в мм над точкой привязки текста, нарисованного последующими командами *TextInstruction*. Используется для генерации подстрочных или надстрочных скриптов.

Applicability: *TextInstruction*

9a-11.2.2.5 Команды перерисовки цвета

Команды перерисовки цвета изменяют цвет условных знаков и пиксельных изображений, нарисованных последующими командами отрисовки.

Таблица 9a-11 – Команды перерисовки цвета

Команда	Параметры	Тип	Начальное состояние	Часть 9	Примечания
OverrideColor	colorToken	string	N/A	9-12.2.2.6 9-12.3.1.2	
	colorTransparency	double	N/A		
	overrideToken	string	N/A		
	overrideTransparency	double	N/A		
OverrideAll	token	string	N/A	9-12.2.2.5 9-12.3.1.1	
	transparency	double	N/A		
ClearOverride					

OverrideColor: *colorToken, colorTransparency, overrideToken, overrideTransparency*

Указывает цвет перерисовки, который должен использоваться для замены исходного цвета в условном знаке или пиксельном изображении, отображаемой с помощью команды отрисовки. Эта команда может выдаваться несколько раз для указания более чем одной замены цветов.

Applicability: *PointInstruction, AreaFillReference, PixmapFill, SymbolFill*

OverrideAll:token,transparency

Заменяет все непрозрачные цвета на заданный цвет. Эта команда заменяет любые команды *OverrideColor*.

Applicability: *PointInstruction, AreaFillReference, PixmapFill, SymbolFill*

ClearOverride

Удаляет все замены цветов.

Applicability: *PointInstruction, AreaFillReference, PixmapFill, SymbolFill*

9a-11.2.2.6 Команды геометрии

Все команды отрисовки, определенные в пункте 9a-11.2.1 ссылаются на геометрию.

Как правило, это геометрия фичера (аналогично пункту 9-11.2.2 Часть 9 *DrawingInstruction::featureReference*). Хост определяет геометрию фичеров, используя ссылку на фичер, предоставляемую при возврате инструкций по отрисовке через *HostPortrayalEmit*, как описано в пункте 9a-14.2.1. Команды геометрии, определенные в настоящем разделе, позволяют изменять нормальное поведение.

Одним из методов переопределения нормального поведения является ограничение команд отрисовки так, чтобы они ссылались либо на отдельные геометрические элементы фичера, либо на любую другую геометрию, определенную в наборе данных (аналогично пункту 9-11.2.2 Части 9 *DrawingInstruction::spatialReference*).

Второй метод переопределения нормального поведения заключается в создании расширенной геометрии (пункт 9-11.1.13 части 9) с использованием команды геометрии. Дополненная геометрия используется, когда в наборе данных отсутствует пространственное изображение. Дополненная геометрия, созданная командой геометрии, будет выполняться последующими командами отрисовки, переписывая геометрию фичеров.

Эта часть не определяет отдельные расширенные инструкции по отрисовке, как в части 9. Вместо этого, все команды отрисовки должны отображаться с использованием расширенной геометрии там, где ее использование возможно.

Чтобы определить геометрию, которую будет выполнять команда отрисовки:

- Если команда дополненной геометрии предшествует команде отрисовки, то следует использовать самую последнюю заданную дополненную геометрию.
- В противном случае, если список пространственных ссылок не является пустым, отрисовка применяется к каждой пространственной ссылке.
- В противном случае должна быть выполнена геометрия фичеров.

Для реализации дополнительных путей хост должен поддерживать список сегментов, в который помещаются команды *Polyline*, *Arc3Points*, *ArcByRadius* и *Annulus*. Этот список поддерживает порядок, в котором создаются геометрии.

Команды геометрии удаляются с помощью команды *ClearGeometry*, которая также очищает список сегментов. Использование *ClearGeometry* позволяет изображению переключаться между рендерингом геометрии фичеров, дополнительной геометрией и пространственными ссылками.

Команды геометрии перечислены в таблице ниже. Тип *point* указывает на передачу пары двойных значений в качестве параметров.

Таблица 9а-12 – Команды геометрии

Команда	Параметры	Тип	Начальное состояние	Часть 9	Прим
SpatialReference	reference	string	-	9-11.2.4	
	forward	boolean	true		
AugmentedPoint	crs	CRSType	-	9-11.2.13	
	x	Point	-		
	y		-		
AugmentedRay	crsDirection	CRSType	-	9-11.2.15	
	direction	double	-		
	crsLength	CRSType	-		
	length	double	-		
AugmentedPath	crsPosition	CRSType	-	9-11.2.16	
	crsAngle	CRSType	-	9-11.2.16	
	crsDistance	CRSType	-	9-11.2.16	
Polyline	point1	Point[]	-	9-12.2.2.11	
	...				
	pointN				
Arc3Points	startPointX	Point	-	9-12.2.2.13	

Команда	Параметры	Тип	Начальное состояние	Часть 9	Прим
	startPointY				
	medianPointX	Point	-		
	medianPointY				
	endPointX	Point	-		
	endPointY				
ArcByRadius	centerX	Point	-	9-12.2.2.14	
	centerY				
	radius	double	-		
	startAngle	double	0		
	angularDistance	double	360		
Annulus	centerX	Point	-	9-12.2.2.15	
	centerY				
	outerRadius	double	-		
	innerRadius	double	outerRadius		
	startAngle	double	0		
	angularDistance	double	360		
ClearGeometry	-	-	-	-	

SpatialReference:reference[,forward]

Указывает ссылку на компоненты пространственного типа фичера, которая определяет геометрию, используемую для отображения следующих команд отрисовки. Не используется, когда вся геометрия фичера должна быть изображена. Каждый раз, когда эта команда вызывается, новая пространственная ссылка добавляется в список пространственных ссылок, который ведет хост. Список пространственных ссылок можно очистить вызовом *ClearGeometry*.

reference Идентификатор пространственного типа, определенный в пункте 13-8 части 13.

forward Если true, пространственный объект используется в том направлении, в котором он хранится в данных. Применяется только к кривым и должен игнорироваться для всех других пространственных типов.

Applicability: Все команды отрисовки.

AugmentedPoint:crs,x,y

Указывает положение. Очищает активные инструкции *AugmentedRay* и *AugmentedPath*.

crs – одна из:

GeographicCRS Географическая CRS с осями широта и долгота, измеренные в градусах.

<i>PortrayalCRS</i>	Декартова система координат с осью У, направленной вверх. Единицы по осям и для расстояний - миллиметры.
<i>LocalCRS</i>	Декартова система координат на базе локальной геометрии. Единицы по осям и для расстояний - миллиметры.
<i>x,y</i>	Координаты точки.

Applicability: *PointInstruction, TextInstruction, NullInstruction*

AugmentedRay: *crsDirection,direction,crsLength,length*

Дополняет геометрию точечного фичера. Указывает линию от положения точечного фичера к другому положению. Положение определяется атрибутами направления и длины. Очищает любые активные инструкции *AugmentedPoint* и *AugmentedPath*.

Если *crsDirection* является *PortrayalCRS* или *LocalCRS*, тогда *crsLength* должна быть *PortrayalCRS* или *LocalCRS*. Аналогично, если *crsLength* является *GeographicCRS*, тогда *crsDirection* должна быть *GeographicCRS*.

crsDirection и *crsLength* – каждая одна из:

<i>GeographicCRS</i>	Углы определяются по часовой стрелке от истинного севера. Расстояние измеряется в метрах.
<i>PortrayalCRS</i>	Декартова система координат с осью У, направленной вверх. Единицы по осям и для расстояний - миллиметры. Углы измеряются в градусах по часовой стрелке от положительной оси У.
<i>LocalCRS</i>	Декартова система координат на базе локальной геометрии. Единицы по осям и для расстояний - миллиметры. Углы измеряются в градусах по часовой стрелке от положительной оси У.

direction Заданное направление луча относительно CRS.

length Длина луча в единицах в зависимости от указанной CRS.

Applicability: *LineInstruction, LineInstructionUnsuppressed, TextInstruction, NullInstruction*

AugmentedPath: *crsPosition,crsAngle,crsDistance*

Указывает хосту собрать все сегменты, ранее созданные командами *Polyline*, *Arc3Points*, *ArcByRadius* и *Annulus*, и сгруппировать их как единую дополненную геометрию. Затем хост должен очистить список сегментов. Очищает все активные инструкции *AugmentedPoint* и *AugmentedRay*.

Для реализации дополненного пути хост должен поддерживать список сегментов. Каждый вызов *Polyline*, *Arc3Points*, *ArcByRadius* и *Annulus* приводит к размещению

геометрии узла в списке сегментов. Эти элементы, выбранные для добавления в список сегментов, определяют расширенный путь.

CRS указывается отдельно для положений, углов и расстояний.

crsPosition, *crsAngle* и *crsDistance* – каждая одна из:

<i>GeographicCRS</i>	Географическая CRS с осями координат широта и долгота, измеряемые в градусах. Углы определяются по часовой стрелке от истинного севера. Расстояния измеряются в метрах.
<i>PortrayalCRS</i>	Декартова система координат с осью У, направленной вверх. Единицы по осям и для расстояний - миллиметры. Углы измеряются в градусах по часовой стрелке от положительной оси У.
<i>LocalCRS</i>	Декартова система координат, построенная на локальной геометрии. Единицы по осям и для расстояний - миллиметры. Углы измеряются в градусах по часовой стрелке от положительной оси У.

Applicability: Все команды отрисовки, кроме *PointInstruction*

Polyline: *positionXstart,positionYstart,positionXto,positionYto[,positionXto,positionYto...]*

Поручает хосту добавить полилинию в список сегментов.

positionXstart,positionYstart,positionXto,positionYto Координаты отрезков полилинии.

Applicability: *AugmentedPath*

Arc3Points: *startPointX,startPointY,medianPointX,medianPointY,endPointX,endPointY*

Поручает хосту добавить дугу, определенную тремя точками, в список сегментов.

startPointX,startPointY Точка, где начинается дуга.

medianPointX,medianPointY Произвольная точка на дуге.

endPointX,endPointY Точка, где заканчивается дуга.

Applicability: *AugmentedPath*

ArcByRadius: *centerX,centerY,radius[,startAngle,angularDistance]*

Поручает узлу добавить дугу, определяемую радиусом, в список сегментов.

centerX,centerY Центр дуги.

<i>radius</i>	Радиус окружности.
<i>startAngle,angularDistance</i>	Сектор, определяющий, где начинается и заканчивается дуга. Если нет, дуга является полным кругом.

Applicability: *AugmentedPath*

Annulus:centerX,centerY,outerRadius[,innerRadius[,startAngle,angularDistance]]

Поручает хосту добавить кольцо в список сегментов. Кольцеобразная область, ограниченная двумя концентрическими кругами. Она может быть ограничена двумя радиусами круга.

Обратите внимание, что наличие параметров *startAngle* и *angularDistance* не означает, что *innerRadius* должен присутствовать. Ниже приведена допустимая команда: `Annulus:0,1,2.34,,56,78`

<i>centerX,centerY</i>	Центр кольца.
<i>outerRadius</i>	Радиус большего круга.
<i>innerRadius</i>	Радиус меньшего круга. Если нет сегмента, то он описывает сектор окружности.
<i>startAngle,angularDistance</i>	Сектор кольцевого сегмента.

Applicability: *AugmentedPath*

ClearGeometry

Очищает все предыдущие команды геометрии и опустошает списки сегментов и пространственных ссылок.

Applicability: *AugmentedPath, SpatialReference*

9a-11.2.2.7 Команды покрытия

Команды покрытия определяют вход в таблицу просмотра, на которую ссылается команда отрисовки *CoverageFill*. Эти команды являются частью функциональности пакета покрытия, описанного в пункте 9-12.7 части 9. Команды покрытия перечислены в таблице 9a-13 ниже.

Таблица 9а-13 - Команды покрытия

Команда	Параметры	Тип	Начальное состояние	Часть 9	Примечания
Command	Parameters	Type	Initial State	Part 9	Notes
NumericAnnotation	decimals	Integer	-	9-12.7.4.4	
	championChoice	ChampionChoice	-		
	buffer	Double	0		
SymbolAnnotation	symbolRef	String	-	9-12.7.4.5	
	rotationAttribute	String	-		
	scaleAttribute	String	-		
	rotationCRS	CRSType	PortrayalCRS		
	rotationOffset	Double	0		
	rotationFactor	Double	1		
	scaleFactor	Double	1		
CoverageColor	startToken	String	-	9-12.7.4.3	
	startTransparency	Double	0		
	endToken	String	-		
	endTransparency	Double	0		
	penWidth	Double	0		
LookupEntry	label	String	-	9-12.7.4.2 1-4.5.3.4	
	lower	Double	-		
	upper	Double	-		
	closure	S100_IntervalType	-		

NumericAnnotation: *decimals, championChoice[,buffer]*

Указывает числовое представление инструкции покрытия. При выполнении команды отрисовки *CoverageFill* числовое значение должно быть отрисовано с использованием текущего шрифта. Однако вместо использования цвета шрифта *FontColor* следует использовать *CoverageColor*.

decimals Число десятичных знаков для отображения в нижнем индексе.

championChoice – один из:

Largest Показывать наибольшее значение в случае несовпадения.

Smallest Показывать наименьшее значение в случае несовпадения.

buffer Буфер, применяемый для обнаружения несовпадений в единицах отображения.

Applicability: *LookupEntry*

SymbolAnnotation: *symbolRef, rotationAttribute, scaleAttribute[, rotationCRS, rotationOffset[, rotationFactor[, scaleFactor]]]*

Указывает символьное представление инструкции покрытия.

<i>symbolRef</i>	Условный знак из Каталога Изображений для рисования.
<i>rotationAttribute</i>	Код атрибута покрытия, используемый для значения поворота условного знака.
<i>scaleAttribute</i>	Код атрибута покрытия, используемый для значения масштабирования размера условного знака.
<i>rotationCRS</i>	Указывает систему координат для вращения.
<i>rotationOffset</i>	Используется для настройки значения <i>rotationAttribute</i> путем сложения перед применением. Это смещение применяется после <i>rotationFactor</i> . Если <i>rotationAttribute</i> не задан, это значение представляет собой значение поворота, применяемое к условному знаку. 0 означает отсутствие корректировки.
<i>rotationFactor</i>	Используется для настройки значения ' <i>rotationAttribute</i> ' путем умножения перед применением. Этот коэффициент применяется до <i>rotationOffset</i> . Значение 1 означает отсутствие корректировки.
<i>scaleFactor</i>	Используется для настройки значения ' <i>scaleAttribute</i> ' путем умножения перед применением. Значение 1 означает отсутствие корректировки.

ПРИМЕР: Предположим, что покрытие имеет атрибуты скорости и направления ветра и изображение представляет собой стрелку, показывающую направление ветра, и длина которой пропорциональна скорости ветра. В этом примере направление ветра указывает направление откуда дует ветер, а нужно показать изображение куда дует ветер. Кроме того, нужно, чтобы скорость ветра 20 узлов была указана путем рисования стрелки в ее нормальном масштабе. В этом случае нужно повернуть стрелку на 180 градусов и масштабировать стрелку на 1/20. Следующие команды могут быть использованы для изображения стрелки:

```
SymbolAnnotation:ARROW,windDirection,windSpeed,PortrayalCRS,180,  
1.0,0.05;  
LookupEntry:Wind,0,360,closedInterval;  
CoverageFill:windDirection
```

Applicability: *LookupEntry*

CoverageColor: *startToken, startTransparency[, endToken, endTransparency][, pen Width]*

Задаёт диапазон цветов, используемый для инструкции покрытия. Если *endToken* и *endTransparency* не указаны, используется один цвет.

startToken, startTransparency Цвет, который следует назначить соответствующему диапазону или использовать в качестве отправной точки в цветовой рампе, когда *'endColor'* определен.

endToken, endTransparency Если указан, цвет, который следует использовать в качестве точки остановки в цветной рампе. Диапазон значений распределяется линейно по диапазону цветов от *'startColor'* до *'endColor'* для получения градиентного эффекта.

penWidth Ширина пера для отображения цвета при отрисовке дискретных точек.

Applicability: *LookupEntry*

LookupEntry: *label, lower, upper, closure*

Создает элемент поиска для использования одной последующей командой рисования *CoverageFill*. Эта инструкция используется для сопоставления команд *NumericAnnotation*, *SymbolAnnotation* и *CoverageColor* с одной позицией таблицы просмотра.

ПРИМЕЧАНИЕ: Для поддержки ассоциаций оповещений со значениями покрытия, которые отличаются от изображения, предшествование команд *NumericAnnotation*, *SymbolAnnotation* или *CoverageColor* не требуется.

ПРИМЕЧАНИЕ: Последующие команды *LookupEntry* требуют переопределения *NumericAnnotation*, *SymbolAnnotation* и *CoverageColor*; например, состояние других команд покрытия должно быть сброшено после обработки *LookupEntry*.

<i>label</i>	Строка, используемая как метка дисплея или поле легенды.
<i>lower</i>	Меньшее значение диапазона поиска.
<i>upper</i>	Верхнее значение диапазона поиска.
<i>Closure</i>	Интервал закрытия диапазона. См. Часть 1 пункт 1-4.5.3.4.

Applicability: *CoverageFill*

9a-11.2.2.8 Команды времени

Команды времени применяют интервалы времени к нижеследующим командам рисования. Интервал времени описывается датой и временем начала и окончания. Команды времени перечислены в Таблице 9a-14.

Таблица 9а-14 – Команды времени

Команда	Параметры	Тип	Начальное состояние	Часть 9	Примечания
Date	begin	string	$-\infty$	1-4.5.3.10	
	end	string	∞		
Time	begin	string	$-\infty$	1-4.5.3.10	
	end	string	∞		
DateTime	begin	string	$-\infty$	1-4.5.3.10	
	end	string	∞		
TimeValid	closure	S100_IntervalType	-	1-4.5.3.11 9-11.2.2	
ClearTime	-	-	-	-	

Date:[begin],[end]

Указывает дату начала и/или окончания интервала времени. Одно из *begin* или *end* должно присутствовать.

begin Начало интервала. S100_TruncatedDate, см. Часть 1 таблицу 1-2.

end Конец интервала. A S100_TruncatedDate, см. Часть 1 таблицу 1-2.

Applicability: TimeValid

Time:[begin],[end]

Указывает дату начала и/или окончания интервала времени. Одно из *begin* или *end* должно присутствовать.

begin Начало интервала. A Time, см. Часть 1 таблицу 1-2.

end Конец интервала. A Time, см. Часть 1 таблицу 1-2.

Applicability: TimeValid

DateTime:[begin],[end]

Указывает дату и время начала и/или окончания интервала времени. Одно из *begin* или *end* должно присутствовать.

begin Начало интервала. A DateTime, см. Часть 1 таблицу 1-2.

end Конец интервала. A DateTime, см. Часть 1 таблицу 1-2.

Applicability: TimeValid

TimeValid[:closure]

Создает интервал времени, который применяется к нижеследующим командам рисования. Интервалы накапливаются до очищения с помощью команды *ClearTime*.

Если какой-либо интервал совпадает с датой просмотра изображения (или другого соответствующего селектора), то команда рисования и связанная с ней информация (например, оповещения) действительны, в противном случае команда и связанная с ней информация отключаются.

Начальные и/или конечные значения интервала времени задаются предыдущими командами *Date*, *Time* или *DateTime*.

closure Строка, определяющая *S100_IntervalType*. См. часть 1 раздел 1-4.5.3.4. Если не указана, по умолчанию *closedInterval*.

ПРИМЕЧАНИЕ: Одиночное значение интервала определяется как *begin = end* и *closure = closedInterval*.

ПРИМЕЧАНИЕ: Хост должен очистить состояние, заданное предыдущими командами *Date*, *Time* и *DateTime* по завершении этой команды.

Applicability: Все команды отрисовки.

ClearTime

Очищает все накопленные временные интервалы и все состояния, заданные предыдущими командами времени.

Applicability: Все команды времени, все команды отрисовки.

9a-11.2.2.9 Команды оповещений

Команды оповещений связывают информацию оповещений с геометрией инструкций по отрисовке.

Таблица 9a-155 – Команды оповещений

Команда	Параметры	Тип	Начальное состояние	Часть 9	Прим
AlertReference	alertReference	string	-	9-11.2.2	
	plan	string	-	9-11.2.5	
	monitor	string	-		

AlertReference[:alertReference[,plan[,monitor]]]

Указывает информацию оповещений, которую нужно ассоциировать с геометрией инструкций по отрисовке.

<i>alertReference</i>	Ссылка на оповещение в каталоге оповещений. Если не указана, то удаляется любая ранее примененная ссылка на оповещение.
<i>plan</i>	Группа просмотра, для которой назначено оповещение при активном планировании маршрута. Если этот параметр не указан, группа просмотра не назначается, и выделение не может быть отключено.
<i>monitor</i>	Группа просмотра, для которой выделено оповещение при активном мониторинге маршрута. Если этот параметр не указан, группа просмотра не назначается, и выделение не может быть отключено.

Applicability: Все команды отрисовки.

9а-12 Определения условных знаков

Определения условных знаков, описанные в пункте 9-12 Части 9, применяются в модели инструкций по отрисовке (пункт 9а-11.2).

9а-13 Библиотека изображений

Организационная структура библиотеки изображений, определенная в пункте 9-13.2 части 9, не меняется. Содержание XSLT папки "Rules" раздел 9-13.2 части 9 заменяется на файлы сценариев Lua. *FileType:rules*, описанный в части 9 раздел 9-13.3.32 используется для идентификации каждого из файлов сценариев Lua.

9а-14 Специальные функции домена изображений

Изображение Lua является реализацией домена сценариев части 13. Функции, описанные ниже, специфичны для этой области сценариев; они являются специфичными для домена функциями, которые будут использоваться в сочетании со стандартными функциями, описанными в части 13.

9а-14.1 Функции каталога домена изображений

Функции, перечисленные в следующих пунктах, реализованы в файлах правил каталога изображений. Они могут вызываться хостом и дополнять стандартные функции каталога, описанные в части 13.

9а-14.1.1 Логический `PortrayalMain(String[] featureIDs)`

Возвращаемое значение:

true Изображение успешно завершено.

false Изображение было прекращено хостом (хост вернул false из *HostPortrayalEmit*).

Параметры:

featureIDs: String[]

Массив, содержащий идентификаторы фичеров, для которых создаются инструкции по отрисовке. Если этот параметр равен нулю (или отсутствует), то изображение будет генерировать инструкции отрисовки для всех реализаций фичеров в наборе данных.

Примечания:

Эта функция вызывается хостом, чтобы начать процесс отображения для реализации набора данных. Впоследствии сценарии изображения будут многократно вызывать *HostPortrayalEmit*, предоставляя хосту инструкции по отрисовке для каждой изображаемой реализации фичера.

Функция возвращается, как только сценарии отображения запущены для завершения; ошибка отбрасывается; или хост возвращает false из *HostPortrayalEmit*.

При использовании кэша отображения, как описано в пункте 9a-5.2.2.1, хосту нужно проходить только не кэшированные *featureIDs* или идентификаторы, связанные с параметрами контекста, значения которых изменились.

9a-14.1.2 void PortrayalInitializeContextParameters(ContextParameter[]

***contextParameters*)**

Возвращаемое значение:

void

Параметры:

contextParameters: ContextParameter[]

Массив объектов ContextParameter.

Примечания:

Предоставляет сценарии изображения с *j* значением по умолчанию для каждого параметра контекста изображения, определенного в Каталоге изображений. *PortrayalCreateContextParameter* следует использовать для создания каждой записи. Хост отвечает за получение параметров контекста изображения из Каталога изображений.

9a-14.1.3 ContextParameter PortrayalCreateContextParameter(String *contextParameterId*, String *contextParameterType*, String *defaultValue*)

Возвращаемое значение:

ContextParameter хранит *defaultValue* с *contextParameterName*.

Параметры:

contextParameterId: String

ID параметра контекста изображения. Допустимые ID определяются в Каталоге изображений.

contextParameterType: String

Тип параметра контекста изображения. Допустимые значения *Boolean*, *Integer*, *Real*, *Text* и *Date*.

defaultValue: String

The default value for the portrayal context parameter. This value is encoded as described in Part 13 clause 13-8.1.

Примечания:

Создает объект ContextParameter для использования в скриптовом окружении.

9a-14.1.4 void PortrayalSetContextParameter(String *contextParameterId*, String *value*)

Возвращаемое значение:

void

Параметры:

contextParameterId: String

Id параметра контекста изображения.

value: String

Новое значение параметра контекста изображения. Это значение кодируется так, как описано в пункте 13-8.1 части 13.

Примечание:

Позволяет хосту изменять значение параметра контекста изображения. Параметр контекста должен быть создан через *PortrayalInitializeContextParameters* до внесения изменений.

9a-14.2 Специфические функции домена изображений хоста

Хост должен реализовывать функцию, описанную в следующем пункте, чтобы поддерживать изображение. Эта функция вызывается из специфических функций каталога, и дополняет стандартные функции хоста, описанные в части 13.

9a-14.2.1 Boolean HostPortrayalEmit(String *featureID*, String *drawingInstructions*, String *observedParameters*)

Возвращаемое значение:

True продолжает обработку. Механизм изображения будет продолжать обрабатывать реализации фичеров.

False Прекращает обработку сценариев. Дополнительные реализации фичеров не будут обрабатываться механизмом изображений.

Параметры:

featureID: String

Используется хостом для уникальной идентификации реализации фичера.

drawingInstructions: String

Все инструкции по отрисовке, созданные для реализации фичера, определяются *featureID*. Эта строка в формате Data Exchange Format (DEF) как описано в части 13.

observedParameters: String

Контекстные параметры, которые наблюдались во время генерации инструкций по отрисовке этого фичера. Это строка в формате DEF.

Примечания:

Эта функция вызывается из Каталога изображений один раз на реализацию фичера, чтобы предоставить хосту инструкции по отрисовке.

Хост может использовать наблюдаемые параметры контекста для выполнения кэширования инструкций отрисовки.