TRANSNAV
http://www.transnav.eu
the International Journal
on Marine Navigation
and Safety of Sea Transportation
Volume 19
Number 4
December 2025

DOI: 10.12716/1001.19.04.04

The Role of Machine-readable Portrayal Catalogue for Displaying New S-100 Format Data in ECDIS

T. Tomczuk
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: The purpose of this article is to familiarize users of Electronic Nautical Charts (ENC) with one of the key components of the new IHO S-100 format, which is an automatic set of data visualization rules called the Portrayal Catalogue (PC). This article presents the current status of data presentation, the structure of the PC divided into file types, the mechanics of the inside PC attached Alert Catalogue (AC) and it crucial role for alerting mariners, an example data flow diagram using the applied Lua language, Lua language principles and the results of a test displaying an anchorage on an ENC S-101 cell. It also discusses the benefits and main risks for manufacturers and users of Electronic Chart and Display Information System (ECDIS) systems when using a PC in S-100 format.

1 INTRODUCTION

Maritime navigation depends largely on the correct position of a seagoing vessel, and nowadays it is primarily displayed on the ECDIS. If it meets the right conditions, ECDIS can be an equivalent system to conventional nautical charts. As a result, maritime operators have been transferring for years from conventional, paper nautical charts in favor of electronic charts.

From the user's point of view, the core components of the system are the Electronic Navigational Charts (ENC), which stores vector information about every object on the chart. In order to standardize the production of ENC charts, the International Hydrographic Organization (IHO) has designed the S-57 hydrographic data exchange standard. It is currently the electronic data format used by Hydrographic Offices (HO) around the world that are required to comply with it when designing uniform ENC data. In recent years, the IHO has conducted intensive research and testing on a new exchange concept. From 2026, new installations of ECDIS systems based on the S-100 standard will be permitted, and from 2029, the use of ECDIS systems compliant with the S-100 standard will become mandatory for new installations. This new standard will be more technologically up-to-date and will allow for the fusion of maritime dependent data. [1]

2 METHOD AND MATERIALS

In this article, a code review analysis was carried out of the displaying ENC data in the S-101 format. For this purpose, the information needed to display ENC navigation data was examined using the IHO S-101 Portrayal Catalog Ed 2.0.0 from the GitHub repository. In addition, the structure of navigation data display in AC and its applications for alerting mariners were analyzed. The S-101 ENC test cell was created from the S-57 ENC cell using the ESRI S-101 Converter, and further displayed information was analyzed by KHOA S-100 Viewer 1.0.25 and CARIS Easy View 6.1.2 with timing analysis.

3 DATA DISPLAY

3.1 *Current status*

At the moment, maritime navigation data was encoded using the right software and visualized with a fixed paper library of symbols known as the S-52 standard. Each object drawn on the chart had an assigned symbol, definitions, color, characteristics and acronyms. However, the way the publication was constructed did not allow for faster changes and automatic adaptation by manufacturers to subsequent updates of the standard. [2]

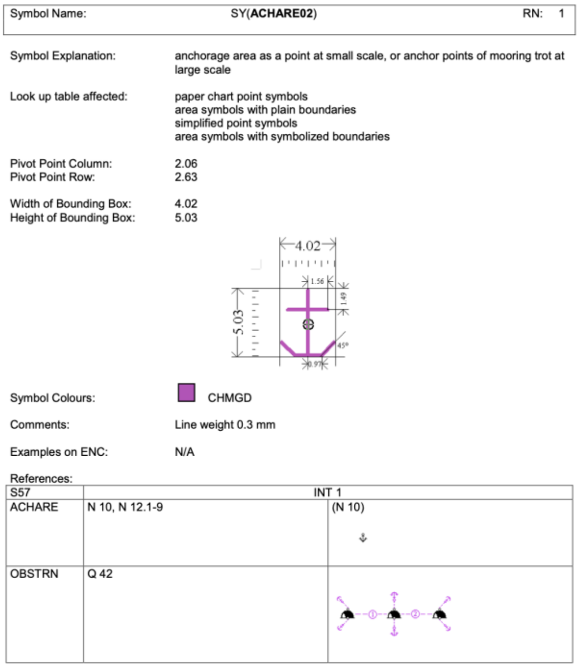

Figure 1. Presentation library for the S-57 standard designated as S-52 Source: IHO S-52 Annex A: IHO ECDIS Presentation Library Ed. 4.0 (3)

3.2 *Data presentation in S-101 format*

In the newly planned S-101 standard, which is based on S-100 format, navigation data is to be defined and presented using two main components: Feature Catalogue (FC) and previously mentioned PC, which will allow for greater data automation and faster updates, enabling effective data and metadata management. At the same time, they will allow work on the standard to continue without having to freeze it for long periods of time. In S-101, FC defines objects, their connections, and how information is encoded, while the PC specifies how data is presented graphically. [3]

3.3 *Construction of Portrayal Catalogue*

PC is a set of machine-readable instructions for graphically representing data in accordance with a specific data model for a particular version of a product specification, [4] They can be created in two ways: manually or using the Portrayal Catalogue Builder (PCB) tool as shown below in Fig. 2. In either case, they must comply with S-100 part 9 and the Portrayal Catalogue Schema to fulfill the validation process. [5]

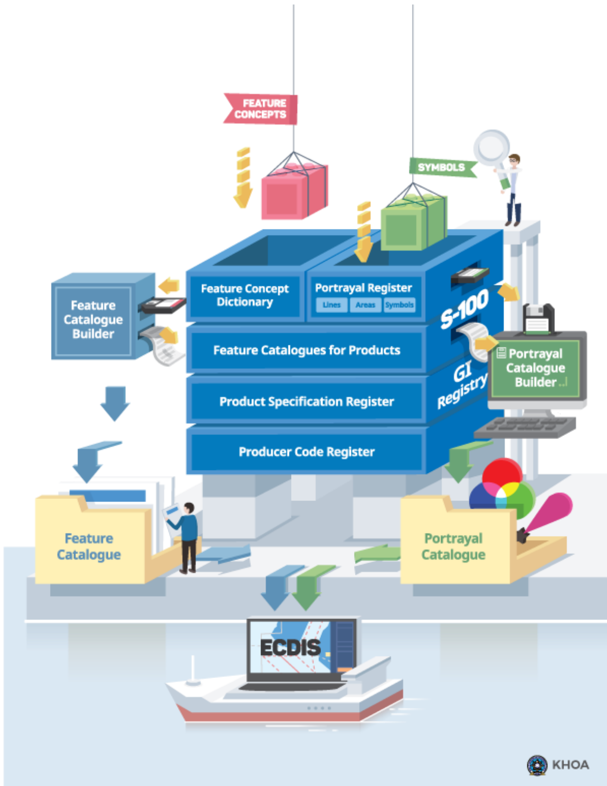

Figure 2. Automatic FC and PC catalogues required for correct data display on ECDIS devices Source: IHO https://iho.int/en/s-100-infrastructure

In its structure, the PC consists of AreaFills .xml files, which are responsible for filling in the appropriate areas, e.g. DQUALA11 describes the CATZOC A1 parameter for the highest measurement accuracy. The colors code used, such as CHMGD corresponding to the dominant magenta color, are defined in the ColorProfiles folder and the ColorProfile .xml file. Line styles are defined under LineStyles, where e.g. ACHARE51 describes the rules for displaying boundary lines for anchorages. Rules in Lua describe the principles of display order and links between relevant elements. Graphic files for each symbol are defined in Symbols as .svg files. In addition, an Alert Catalogue is also included to enable dynamic filtering of alerts to the PC, which is described in the following part of this article. Below in Table 1. is a summary of PC structure described divided into files and their assigned functions. [6]

Table 1. Construction of PC with described different parts and their application

Folder	File type	Function
AreaFills	.xml	Filling in areas
ColorProfiles	.xml	Colors definitions
LineStyles	.xml	Line styles
Rules	.Lua	Display and sequence
Symbols	.svg	Presentation of symbols
Alert Catalogue	.xml	Dynamic alert catalogue

3.4 *Alert Catalogue*

The new S-100 standard, on which the ENC S-101 data format for the PC is based, also includes the AC. It contains entries corresponding to individual alerts and allows these entries to be linked to the presented rules in PC. This allows the ECDIS system to generate files based on spatial data, as there are many similarities between generating alerts and presenting features. While the PC is designed to process an encoded data set into drawing instructions, the AC is designed to translate an encoded data set into alert instructions. These instructions specify which spatial elements from the encoded data set should be evaluated by the alert processing implemented in the ECDIS system, and provide the ECDIS system with the information it needs to generate an alert. ECDIS system manufactures (host) will be responsible for implementing the spatial evaluation. Any changes to the AC will be managed in the same way as changes to the PC, allowing ECDIS manufacturers to update only machine-readable files instead of updating the entire software. This solution will allow producers to adapt more effectively to updates of the S-100 standard.

Table 2. Detailed description of alarm types, definitions and their areas of application

Alarm type (Section)	Explanation	Appearance in AC
Alarm (11.4.3)	Pass closer than set distance from the safety contour	Appears in AC, alert id = "SafetyContour"
Warning or Caution, or Indication (11.4.4)	Pass closer than set distance from an area with special conditions	Appears in AC, alert id = "ProhAre"
Warning or Caution, or Indication (11.4.6)	Pass closer than set distance from a danger in route monitoring mode	Appears in AC, alert id = "NavHazard"
Alarm (11.4.5)	Deviation from route	Currently not listed in AC
Warning (11.4.11-13)	Positioning system failure, Approach to critical point, Different geodetic datum	Not available in AC, implementation on the ECDIS manufacturer side (host)
Warning or Indication (13.2)	Malfunction of ECDIS	Not available in AC, implementation on the ECDIS manufacturer side (host)
Indication (5.8.3, 6.1.1-3, 7.3, 8.5, 10.5, 13.1)	Default safety contour, Information overscale, Larger scale ENC available, Information not displayed owing to scale minimum, Different reference system, No ENC available, Customized display, System test failure	Not available in AC, implementation on the ECDIS manufacturer side (host)
Indication (11.3.6-7, 11.4.7-8)	Route planning closer than set distance from the safety contour, Route planning closer than set distance specified area or danger, Monitored route pass closer than set distance from the safety contour, Monitored route pass closer than set distance from a specified area or danger	Appears in AC, alert id = "SafetyContour" and alert id = "ProhAre" as part of RteMonitor and RtePlan

The alarm model included in the AC, is part of the PC and complies with resolution MSC.530(106)/REV.1 of 24 May 2024. [7] It is presented as follows: all alarms are further grouped into four categories (alarm, warning, caution, indication), the required user settings for the distance parameters before the Safety Contour, a given area, or a hazard have been formally described, some of the alerts will be on the PC side (due to their association to features encoded on ENC charts), while others will remain on the side of ECDIS producers. It is a main difference from previous solutions. The prioritization of alarms will also depend on the route planning or monitoring mode. [8] A detailed distinction is presented below in Table 2. [9]

3.5 *Lua*

Rules uses the Lua language, which is a lightweight programming language that allows for normal procedural programming while also describing data. Lua is commonly used as a comprehensive, extensible scripting language. As such, there is no such thing as a 'main program' in Lua; it is not a language designed for writing applications from scratch, but rather as a supplement. Lua is activated within the host program (written in C or C++ for example), which can call a function to run fragments of the program written in Lua as planned in the case of PC, where the language is used to process visualization rules for individual objects on ENC charts.

The following keywords are reserved in Lua: *and, break, do, else, elseif, end, false, for, function, if, in, local, nil, not, or, repeat, return, then, true, until, while*. In the PC used, conditional statements such as *if, else, elseif, return, end*. The *if-then-end* structure in Lua works as follows:

1. *if* <condition> *then*: checks whether, <condition> is true; only *false* and *nil* values are considered as false - any other value (including '0' or an empty string) is treated as true.
2. *if* <condition> is accepted as true, the block of instruction between *then*, *elseif*, *else end* is executed.
3. *elseif* <other condition> *then* (optional): if the first condition (*if*) is false, Lua checks the condition written after elseif. In this case, you can use multiple elseif in sequence, one after another.
4. *else* (optional): if none of the previously described conditions (from *if*, *elseif*) are accepted as true, the code block after the word else is executed.
5. *end*: this keyword is mandatory and defines the end of the entire conditional statement *if*. [10]

3.6 *Data visualization flow diagram*

The presentation of ENC data in the new S-100 format using the Lua language can be carried out as shown in Fig. 3 below. First, the process begins in the host environment (e.g. in the ECDIS system), which creates a list of IDs for all features to be displayed in the system. This list is then passed to the Lua engine for processing according to the Portrayal Rules stored in .Lua files. For each identifier on the list, Lua scripts use the S-100 Portrayal API to access additional information about the object. The Portrayal API communicates with host data using the Host Integration API. Thanks to this process, scripts can send queries for specific data without having to load the entire data set at once. After receiving the relevant feedback, the Lua script for a given object executes the visualization rules and emits a set of Draw Instruction, which are returned to the host individually via the Host Integration API. Finally, the host collects all generated drawing instructions into a single DisplayList. After processing all objects, the host executes in this list, which results in the data being displayed on the system screen. In this proposed solution, the Lua approach proves to be more efficient than XSLT technology. [11]

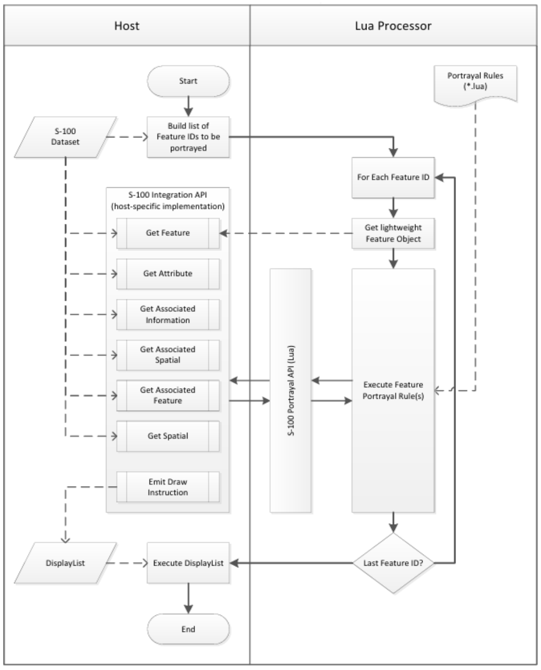

Figure 3. The general architectural pattern proposed by the IHO for ENC data visualization flow for S-100 Source: IHO: Paper for Consideration by the S-100 Working Group S-100 Portrayal Support for Lua

3.7 *Example of drawing an anchorage*

In the example below, the Lua language instructions if, elseif, else, end create decision logic that selects the appropriate set of instructions for drawing an anchorage on an ENC chart based on the object type, display settings and attributes. For example, for AnchorageArea the visualization rule attached as Annex A, structure is as follows:

1. *if* feature.PrimitiveType == PrimitiveType.Point *then ...*
   The interpreter first checks whether the geometric type of the object (*feature.PrimitiveType*) is point geometry (*Point*). If this is true, it executes the code inside this block (draw the *ACHARE02* symbol corresponding to the point anchorage from the Symbols package and draws possible text), then skips the remaining conditions (*elseif, else*) and goes to the end of the function (*end*).
2. *elseif feature.PrimitiveType == PrimitiveType.Surface and contextParameters.PlainBoundaries then ...*
   If the object turns out to have no point geometry, the Lua language moves on to this condition and verifies two components simultaneously (using *and* operator, which requires both conditions to be true):
   – whether the object type is a *Surface*,
   – whether the boundary display mode *PlainBoundaries* is enabled.

   If both conditions are met, the interpreter enters this block, where it encounters another nested *if-else-end* statement.
3. *elseif feature.PrimitiveType == PrimitiveType.Surface then ...*
   This condition is triggered only when conditions (1) and (2) above are false. In this case, the object is a surface, but the *PlainBoundaries* mode is disabled. Here, there is also a nested *if-else-end* statement that selects the appropriate symbols and line styles depending on the anchorage category. [12]
4. The *else* block will only be executed if none of the above conditions were true - that is, if the anchorage object is neither a point nor a surface, which of course cannot happen in the real world. In this situation, the script will report an error using the *error()* function.

4 RESULTS

In the example below in Fig. 4, an image of an anchorage has been generated in Plain Boundaries and Symbolized Boundaries modes for the ENC S-101 101PL0045150000.000 test cell converted from S-57 using the ESRI S-101 Converter tool.

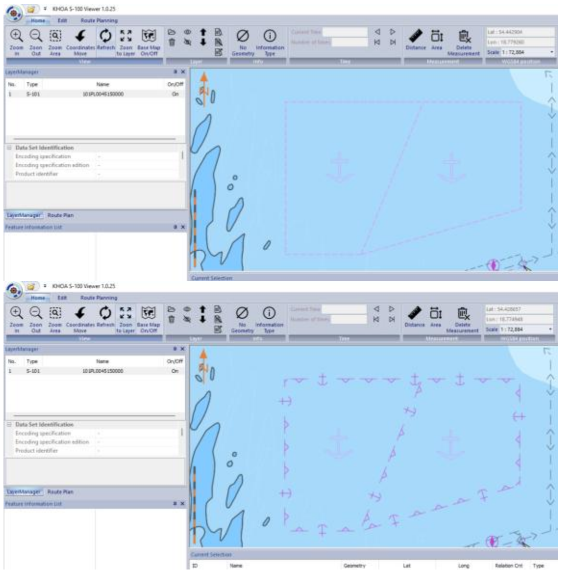

Figure 4 View of the drawn anchorage in PlainBoundaries mode (top) and SymbolizedBoundaries (bottom) in KHOA S-100 Viewer 1.0.25 software with PC 2.0.0 attached.

Two types of software were used to generate the anchorage view: KHOA S-100 Viewer and CARIS Easy View. The summary results with timing analysis are shown in Table 3. Each of them used a different image rendering technique. The method of generating data using a PC and the Lua language proved to be more time-efficient than XSLT, the same applies to changing the data view setting the Safety Contours and Display Mode parameters.

Table 3. Time of rendering ENC S-101 cell depending on different software

Date	ENC ID	RenderSoftware	Result
13-10-2025	101PL00451500000PC	KHOA S-100 Viewer 1.0.25	0.5s
13-10-2025	101PL00451500000XSLT	CARIS Easy View 6.1.2	1.5s

5 DISCUSSION

The machine-readable PC is a powerful tool for hydrography and GIS environments, and its implementation offers significant benefits but also potential risks.

The biggest advantage of an automatic data visualization catalogue is that it ensures that all ECDIS systems interpret data in the same way, thereby reducing any errors in symbolization and accelerating work. In the future, when an update is needed, the PC can be updated automatically without modifying individual visualization rules. The update work can be done 'in the background', without freezing standards for long periods of time. This approach shortens the implementation cycle for new versions, which will be continuously adapted to current technology. In addition, common chart rendering rules allow for integrated multi-layer charts, enabling interoperability and the combination of multiple S-100 products. Formally, a PC has an XML structure and can be automatically validated against XSD rules. This approach allows for quality control and catalogue compliance.

The challenge lies in its technical complexity - the PC catalogue is based on complex relationships and data flows, and even the smallest errors in the catalogue, such as a typo or an incorrect *namespace* [14] can prevent the image from being generated correctly in the ECDIS systems. The same applies to XSD schemas, if they are incorrect or incorrectly selected, all products may be misinterpreted. Another threat may be a deliberate minor change in the entire PC, followed by automatic distribution to recipients. Thousands of schemas use the wrong rule, resulting in quality degradation, misinterpretation and in extreme cases, navigational risk.

6 CONCLUSIONS

Despite these risks, automation and PC significantly accelerates work and improves final product quality, but this must go hand in hand with testing, continuous monitoring and change control processes, especially in operational environments. If this is ensured, the benefits of a machine-readable catalogue far outweigh the potential risks. [15]

ACKNOWLEDGMENT

The herein study has been supported by Gdynia Maritime University internal grant #WN/2025/PZ/01

REFERENCES

[1] International Hydrographic Organization (2024) S-100 Roadmap for the Implementation Decade (2020-2030, v4.0 Available online: https://iho.int/uploads/user/About%20IHO/Council/S-100_ImplementationStrategy/S100_Roadmap_Decade_v4.0_clean_October2024.pdf. (accesed on 01 Sep 2025)

[2] International Hydrographic Organization (2020) S-52: Annex A — IHO ECDIS Presentation Library, Edition 4.0(.3). Available online: https://iho.int/uploads/user/pubs/standards/s-52/S-52%20PresLib%20Ed%204.0.3%20Part%20I%20Addendum_Clean.pdf (accessed on 02 Sep 2025).

[3] H. S. Na, Y-S. Choi, M. S. Kim, S. R. Lee, and D.U Kim (2025) Assessment of S-101 Electronic Navigational Chart Accuracy and Reliability through Validation, Sensors and Materials, Vol. 37, No. 2 727–743. https://doi.org/10.18494/SAM5453

[4] H.S Choi, D.W Kang, S.W Oh, Y.J Kim (2021) A Study on Development of Machine-readable Platform for S-100 ECDIS, Journal of Navigation and Port Research Vol 45(2) 61-68.

[5] International Hydrographic Organization (2025) S-100: Universal Hydrographic Data Model — Infrastructure — Portrayal Catalogue Builder. Available online: https://iho.int/en/portrayal-catalogue-builder (accessed: 03 Sep 2025).

[6] International Hydrographic Organization (2025), S-101_Portrayal-Catalogue — PortrayalCatalogue 2.0.0. Available online: https://github.com/iho-ohi/S-101_Portrayal-Catalogue/tree/main/PortrayalCatalog. (accessed on 03 Sep 2025)

[7] A. Weintrit (2022) Revision of the IMO's Performance Standards for ECDIS. Three Versions of Performance Standards in Use, Transav Vol 16, No. 4 675-683. http://dx.doi.org/10.12716/1001.16.04.09 10.12716/1001.16.04.09

[8] International Hydrographic Organization (2019), Paper for Consideration by the S-100 TSM — Proposed Alerts and Indications Model for S-100. Available online: https://legacy.iho.int/mtg_docs/com_wg/S-100WG/TSM7/TSM7_2019_5.2_Alerts.pdf (accessed on 10 Sep 2025).

[9] International Maritime Organization (2024), Resolution MSC.530(106)/Rev.1 — Performance Standards for Electronic Chart Display and Information Systems (ECDIS), Annex 18, adopted 24 May 2024. Available online: (https://www.mardep.gov.hk/filemanager/en/share/msnote/pdf/msin2446anx1.pdf) (accessed on 16 Sep 2025).

[10] Lua.org, (2025) Lua 5.1 Reference Manual. Available online: https://lua.org.pl/5.1/manual.html#1. (accessed on 16 Sep 2025).

[11] International Hydrographic Organization (2017), Paper for Consideration by the S-100 Working Group — S-100 Portrayal Support for Lua. Available online: https://legacy.iho.int/mtg_docs/com_wg/S-100WG/S-100WG2/S100WG02-10.8_S-100_PortrayalSupport_for_Lua.pdf. (accessed on 17 Sep 2025).

[12] E. Caroletti (2021) ECDIS Plain vs Symbolised line styles in ENC. Available online: https://www.linkedin.com/pulse/ecdis-plain-vs-symbolised-line-styles-enc-emiliano-caroletti- (accessed on 29 Sep 2025).

[13] International Hydrographic Organization (2025) S-101 Portrayal Catalogue 2.0.0 — Rules — AnchorageArea.lua. Available online: https://github.com/iho-ohi/S-101_Portrayal-Catalogue/blob/main/PortrayalCatalog/Rules/AnchorageArea.lua (accessed on 17 Sep 2025).

[14] International Hydrographic Organization (2022), Paper for Consideration by the S-100WG — Schema use in Presentation Library XML and XSD files. Available online: https://iho.int/uploads/user/Services%20and%20Standards/S-100WG/S-100WG7/S100WG7-4.19_2022_EN_Schema%20Use%20in%20Presentation%20Library%20XML%20and%20XSD%20files.pdf (accessed on 07 Oct 2025).

[15] International Hydrographic Organization (2022), Electronic Navigational Chart (ENC) Product Specification, Edition 1.1.0. Available online: https://iho.int/uploads/user/Services%20and%20Standar

ds/S-100WG/S-100WG7/S-101%20ENC_Product_Specification_1.1.0.20221208_Redline.pdf (accessed on 6 Oct 2025).

ANNEX A

Rule in Lua for Anchorage Area

```
--
-- ISSUES: {PSWG #72, PC #121}, {PSWG #28, PC #124}
-- #169
-- #184
-- #206

-- Referenced portrayal rules.
require 'RESTRN01'

-- Anchorage area main entry point.
function AnchorageArea(feature, featurePortrayal, contextParameters)
    local viewingGroup

    local COA = feature.categoryOfAnchorage

    featurePortrayal:AddInstructions('AlertReference:ProhAre,53023,53023')

    if feature.PrimitiveType == PrimitiveType.Point then
        -- Simplified and paper chart points use the same symbolization
        viewingGroup = 26220
        if contextParameters.RadarOverlay then
    featurePortrayal:AddInstructions('ViewingGroup:26220;DrawingPriority:18;DisplayPlane:OverRadar')
        else
    featurePortrayal:AddInstructions('ViewingGroup:26220;DrawingPriority:18;DisplayPlane:UnderRadar')
        end

        if contains (15, COA) then     -- 15 - Reported Anchorage PSWG #72
    featurePortrayal:AddInstructions('PointInstruction:ACHARE03')
        else
    featurePortrayal:AddInstructions('PointInstruction:ACHARE02')
        end

        if feature.featureName[1] and feature.featureName[1].name then
            featurePortrayal:AddInstructions('LocalOffset:-3.51,-7.02;FontSize:10;FontColor:CHBLK')
    featurePortrayal:AddTextInstruction(EncodeString(GetFeatureName(feature, contextParameters)), 26, 24, 26220, 18)
        end
    elseif feature.PrimitiveType == PrimitiveType.Surface and contextParameters.PlainBoundaries then
        if contains(15, COA) then -- #169 reported anchorage must have point geometry
            error('Surface is not a valid geometry for reported anchorages')
        else
            viewingGroup = 26220
    featurePortrayal:AddInstructions('ViewingGroup:26220;DrawingPriority:9;DisplayPlane:UnderRadar')
    featurePortrayal:AddInstructions('PointInstruction:ACHARE51')
            featurePortrayal:SimpleLineStyle('dash',0.64,'CHMGF')
    featurePortrayal:AddInstructions('LineInstruction:_simple_')

            if feature.featureName[1] and feature.featureName[1].name then
                featurePortrayal:AddInstructions('LocalOffset:-3.51,-7.02;TextAlignHorizontal:End;FontSize:10;FontColor:CHBLK')
    featurePortrayal:AddTextInstruction(EncodeString(GetFeatureName(feature, contextParameters)), 26, 24, 26220, 9)
            end
            RESTRN01(feature, featurePortrayal, contextParameters, viewingGroup)
        end
    elseif feature.PrimitiveType == PrimitiveType.Surface then
        if contains(15, COA) then -- #169 reported anchorage must have point geometry
            error('Surface is not a valid geometry for reported anchorages')
        else
            viewingGroup = 26220
    featurePortrayal:AddInstructions('ViewingGroup:26220;DrawingPriority:9;DisplayPlane:UnderRadar')
    featurePortrayal:AddInstructions('PointInstruction:ACHARE51')
    featurePortrayal:AddInstructions('LineInstruction:ACHARE51')

            if feature.featureName[1] and feature.featureName[1].name then
                featurePortrayal:AddInstructions('LocalOffset:-3.51,7.02;FontSize:10;FontColor:CHBLK')
    featurePortrayal:AddTextInstruction(EncodeString(GetFeatureName(feature, contextParameters)), 26, 24, 26220, 9)
            end

            RESTRN01(feature, featurePortrayal, contextParameters, viewingGroup)
        end
    else
        error('Invalid primitive type or mariner settings passed to portrayal')
    end

    return viewingGroup
end
```