# Sentinel-1 IPF Auxiliary Product Specification

Prepared By: Matthieu Bourbigot, Pauline Vincent

Harald Johnsen, Riccardo Piantanida

Checked By: Guillaume Hajduch

Quality Assurance: Julie Poullaouec

Project Manager: Guillaume Hajduch

(signature / date)

**Document Number: S1-RS-MDA-52-7443**
**S-1 MPC Nomenclature: DI-MPC-PB**
**S-1 MPC Reference: MPC-0241**

# CHANGE RECORD

From issue 1.0 to 2.10, the Sentinel-1 IPF Auxiliary File Specification was maintained by a consortium led by MDA under the reference S1-RS-MDA-52-7443.

The S-1 IPF and associated documentation is then maintained by the S-1 Mission Performance Center. From the issue 3.0 the Sentinel-1 IPF Auxiliary File Specification is maintained by the S-1 Mission Performance Center which is a consortium led by CLS.

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| 1/0 | June 26, 2009 | All | First Issue. |
| 2/0 | July 30, 2010 | All | Second Issue.<br>Updated based on content of CCN N.2<br>Addressed PDR L1 RIDs (High):<br>RID-2 & 17: Re-organized the layout of Sections 4.4, 5.4, 6.4.<br>RID-5: Added a description of the manifest file as Appendix B.<br>RID-19: Added a note to Section 4.4 to explain that AUX_PAR fields are not polarisation dependent.<br>RID-22(2): Use of the word “type” has been removed from all element names.<br>RID-24: Quick-look and Browse processing parameters structures have been updated according to the algorithm document.<br>RID-25: Various field descriptions have been clarified and elaborated.<br>RID-28: Referenced $C_0$ parameter from S-1 C&C Plan. $C_0$ is included in the absoluteCalibrationConstant.<br>RID-29 & 44: Descriptions of threshold parameters have been updated.<br>RID-38: Swath identifier has been added to the radarSamplingRate record.<br>RID-39: Azimuth steering angles have been added to the AUX_INS file.<br>RID-48: Antenna elevation pattern parameters and descriptions have been updated.<br>Addressed PDR L1 RIDs (Medium/Low): |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | | RID-1, RID-4, RID7, RID-8, RID-10, RID-11, RID-12, RID-14, RID-16, RID-18, RID-30, RID-32, RID-33, RID-40, RID-45, RID-49 |
| 2/1 | Nov. 10, 2010 | All | Second Issue, First Revision<br>Addressed Major Delta PDR L1/PDR L2 RIDs:<br>S1IPFPDR-92: Split L1 and L2 processor parameters into separate files.<br>S1IPFPDR-107: Removed slicing-related parameters that belong in the internal parameters file.<br>S1IPFPDR-111: Updated the replica reconstruction as per Issue/Revision 1/2 of the Algorithm Document.<br>S1IPFPDR-116: Consolidated and reorganized the L2 OCN processor parameters.<br>S1IPFPDR-119: Split the cell size parameter for the OWI grid into range and azimuth.<br>S1IPFPDR-123: Updated time line selection to use ECC number rather than instrument mode.<br>S1IPFPDR-130: Added SWST bias parameter as per Issue/Revision 1/2 of the Algorithm Document.<br>S1IPFPDR-132: Coastlines auxiliary data has been removed from the specification.<br>S1IPFPDR-151: Updated applicable documents for SAFE 1.2.<br>S1IPFPDR-152: Consolidated auxiliary data list with ICD and PD.<br>Addressed Minor Delta PDR L1/PDR L2 RIDs:<br>S1IPFPDR-93, S1IPFPDR-95, S1IPFPDR-97, S1IPFPDR-99, S1IPFPDR-102, S1IPFPDR-106, S1IPFPDR-113, S1IPFPDR-117, S1IPFPDR-118, S1IPFPDR-120, S1IPFPDR-122, S1IPFPDR-124, S1IPFPDR-127, S1IPFPDR-131, S1IPFPDR-133, S1IPFPDR-155, S1IPFPDR-158 |
| 2/2 | May 06, 2011 | All | Second Issue, Second Revision<br>Changed “length” attribute to “count”.<br>Addressed Major Delta PDR L2 RIDs:<br>S1IPFPDR-157: Updated description of auxiliary file manifest in Appendix B.<br>S1IPFDPDRL2-23: Reviewed AUX_PP2 parameters |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | | for RVL component (note that most RVL parameters will be specified in internal parameter files). |
| | | | S1IPFDPDRL2-27: Updated DEM description. |
| | | | S1IPFDPDRL2-30: Addressed S1IPFPDR-96 and S1IPFPDR-101 RIDs from PDR L2. |
| | | | Addressed Minor Delta PDR L2 RIDs: |
| | | | S1IPFPDR-96 |
| | | | S1IPFDPDRL2-24, S1IPFDPDRL2-25, S1IPFDPDRL2-26, S1IPFDPDRL2-29 |
| | | | The following RIDs were raised but no update to this document were required: |
| | | | S1IPFPDR-101: No auxiliary information is required to determine the UTC/GPS time relationship, as this is derived from the input L0 product. No leap second information needs to be provided to the IPF if this UTC/GPS relationship is available. |
| | | | S1IPFDPDRL2-20: Refers to updates to other documents. |
| | | | S1IPFDPDRL2-28 |
| 2/3 | Sep. 19, 2011 | | Second Issue, Third Revision |
| | | | Addressed Major CDR L1&L2 RIDs: |
| | | Section 3 | IPFCDR-28: Updated Sec. 3 to clarify which auxiliary files are SAFE-wrapped and which are not. The DEM was removed as it is no longer an auxiliary product. |
| | | Section 6.3 | IPFCDR-35: The AUX_INS channelParams was renamed internalCalibrationParams, and the fields were updated to align with the updates to the algorithm document internal calibration section. The description of the pgProductModel was updated to be relative to the ANX time. |
| | | Section 6.3.3 | IPFCDR-121: Added default noise to AUX_INS internalCalibrationParams. |
| | | | Addressed Minor CDR L1&L2 RIDs: |
| | | All | IPFCDR-29, IPFCDR-30, IPFCDR-31, IPFCDR-32, IPFCDR-33, IPFCDR-34, IPFCDR-36, IPFCDR-86, IPFCDR-126 |
| 2/4 | May 01, 2012 | | Second Issue, Fourth Revision |
| | | Section 4 | AUX_PP1 Updates: |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | | Changed SNR parameters in PP1 to RMS error. |
| | | | Changed type of instantaneousBandwidth. |
| | | | Removed applyApplicationLut field. |
| | | | Moved gain from postProcParams to slcProcParams. |
| | | | Added applyRangeSpreadingLossFlag to slcProcParams. |
| | | | Added aziProcBlockParams to commonProcParams. |
| | | | Removed applyReplicaWhiteningFlag. |
| | | | Moved output mean/stddev expected/threshold fields from slcProcParams to commonProcParams. |
| | | | Removed references to “Browse” products. |
| | | | Clarified use of the terrain height parameters within the preProcParams. |
| | | | Changed the units of maxXCorrPulseIrw to samples. |
| | | | Removed the units from maxPgAmpError and changed the units of maxPgPhaseError to radians. |
| | | Section 5 | AUX_CAL Updates: |
| | | | Changed the minimum cardinality of calibrationParams from 58 to 60. |
| | | Section 6 | AUX_INS Updates: |
| | | | Added deltaTGuard1, deltaTDecFilter, numSilentPriPre/Post. |
| | | | Renamed azimuthSteeringAngles to azimuthSteeringRate and changed the definition. |
| | | Section 7 | Minor updates to AUX_PP2 fields. |
| | | Section 8 | Updated AUX_SCS description, name and definition. |
| | | Section 13 | Updated AUX_ICE description and data file name. |
| | | App B | Added auxProductType to manifest. |
| 2/5 | Aug. 27, 2012 | | Second Issue, Fifth Revision |
| | | 4-10, 4-19 | Removed taperBwExtentFraction field from AUX_PP1. |
| | | 4-19 | Changed nominalBeamWidth units from radians to degrees in AUX_PP1. |
| | | 5-5 | Changed kNoise to noiseCalibrationFactor in AUX_CAL. |
| | | 6-7 | Added units for azimuthSteeringRate in AUX_INS. |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | 3-5, 8-1 | Updated naming convention for AUX_SCS. |
| 2/6 | Feb. 18, 2013 | | Second Issue, Sixth Revision |
| | | | Updates for CR#4: |
| | | Sections 3.2.2, 3.4 | Added support for notch modes and RF calibration mode. |
| | | Table 3-5 | S1IPFOSAT-1: Corrected the range of values for dcMethodType enum. |
| | | | Removed swathNumberType.<br>Added signalType.<br>Added topsFilterOriginType.<br>Added floatCoefficientArray and floatPatternArray types. |
| | | Table 4-5 | Added correctRxVariationFlag and topsFilterOrigin fields. |
| | | Table 4-9 | Added performInternalCalibrationFlag field. |
| | | Table 4-11 | Changed dcPredefinedCoefficients to use new floatCoefficientArray type. |
| | | Sections 5, 6 | Removed swathNumber fields. Updated the descriptions to indicate that the swath is now used as the index. Updated the cardinality of all affected lists. |
| | | Section 6.1 | Added rxGainCorrectionParams to AUX_INS. |
| | | Sections 3.4, 6.1 | Removed numSilentPriPre and numSilentPriPost fields from timeline definition in AUX_INS, and added a Silent singalType option. |
| | | Section 6.3.2 | Added parameter record definitions for the Rx variation correction. |
| | | Table 6-7 | Changed replica coefficient fields to use new floatCoefficientArray type. |
| | | Table 6-18 | Increased cardinality of isp field. |
| | | Sections 11, 12, 13 | S1IPFOSAT-3: Updated naming standard for AUX_WND, AUX_WAV and AUX_ICE data file components. |
| | | | Updated size estimates based on existing auxiliary product sizes. |
| | | Figures 4-2, 4-3, 5-1, 6-3, 6-4, 6-5 | Updated figures based on auxiliary file content changes. |
| 2/7 | Mar. 15, 2013 | | Second Issue, Seventh Revision |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | Table 3-5 | Added swathNumberType. |
| | | Section 6.3.4 | Added swath map to the AUX_INS file. |
| | | Figure 6-5 | Updated figure based on auxiliary file content changes. |
| 2/8 | Jan. 30, 2014 | | Second Issue, Eighth Revision |
| | | All | Removed copyright, use and disclosure notice. |
| | | Table 3-5 | Added new primitive types: bandwidthType, calCombinationMethodType, rxPolarisationType, doubleCoefficientArray. |
| | | | Updated length of floatCoefficientArray to support Rx Gain Correction. |
| | | Table 4-5 | Added a new field orbitModelMargin to the common processing parameters. |
| | | Table 4-8 | Clarified the use of the aziProcBandwidth parameter. |
| | | | Clarified that maxFdc is *maxDeltaFdc* from the algorithm definition. |
| | | Table 4-11 | Corrected description of dcRmsErrorThreshold. |
| | | Table 4-14 | Clarified the meaning and use of the nominalBeamWidth parameter. |
| | | Table 4-15 | Clarified that the annotation vector step size parameter is used for the calibration and noise vectors. |
| | | | Clarified the meaning of the createInternalSLCFlag parameter. |
| | | Table 5-5 | Clarified that beam nominal range parameters are not used within the IPF. |
| | | Figure 4-2<br>Figure 6-1<br>Figure 6-3<br>Figure 6-4<br>Figure 6-5<br>Figure 6-6 | Updated figures based on format changes. |
| | | Table 6-2 | Added deltaTSuppr parameter for consistency with deltaTGuard1. |
| | | Table 6-5 | Update description of pulseParams record and removed electronicLosses. |
| | | Table 6-7 | Update description of the coefficient arrays to clarify that they apply to the imaging replicas and added the formulae used to calculate them. |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | | Added new nominalTxPulseLength parameter for configuring the Spurious Signal Filtering operation in the IPF. |
| | | p. 6-9 | Removed electronicLossesType. |
| | | Table 6-8 | Updated description to remove references to phase correction. |
| | | Table 6-9 | Update description for use of the coefficient arrays and removed the gainOffsetCoefficients and phaseCoefficients. |
| | | Table 6-11 | Clarified that the imaginary (phase) part of the extractedGain parameter shall always be set to 0. |
| | | | Clarified the meaning and dimension of $PG_{ref}$. |
| | | | Removed deltaTDecFilter. |
| | | | Added replica and PG PCC parameters. |
| | | Table 6-12 | Clarified the use and dimension of the values within the PG model. |
| | | p. 6-17 | Added Tables 6-13 and 6-14 for PCC parameter type definitions. |
| | | Table 6-20 | Added bandwidth parameter. |
| | | Table 6-23 | Removed dValuesLutList. |
| | | | Added thresholdLutList, tguLut and tileLut. |
| | | Section 6.3.5 | Removed dValuesLutListType and dValuesLutType. |
| | | p. 6-32 | Added Tables 6-28 and 6-29 for raw data decoding threshold parameter type definitions. |
| | | Section B2.1 | Updated namespaces for wrapped metadata elements. |
| 2/9 | May 30, 2014 | | Second Issue, Ninth Revision |
| | | Table 4-8 | Changed the data type and description of maxFdc so that it applicable to TOPS. |
| | | Table 4-14 | Changed the data type and description of gain so that it is set in a polarisation-specific way. |
| | | Table 6-11 | Clarified the description of the timeDelay field. |
| 2/10 | Nov. 21, 2014 | | Second Issue, Tenth Revision |
| | | Table 5-3 | Clarified that the absolute calibration constant must be the same for all swaths and polarisations within a mode. |
| | | Table 5-5 | Complex elevation antenna pattern. |
| 3/0 | July 22, 2015 | | Third Issue |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | Table 7-9 | Changed description for gmfIndex and prIndex parameters |
| 3/1 | Feb. 29, 2016 | | Third Issue, First Revision |
| | | Table 7-9 | [MPCS-1076]Introduce a new variable input "activateNoiseCorretion" on the AUX_PP2 for the noise correction for future IPF versions |
| | | Table 5-5 | [IPF-151]: Updated EAP correction formula |
| 3/2 | Sep. 12, 2016 | Appendix 6 | [IPF-189] link to S1__AUX_WND file naming convention in [S1-AD-18]-S1-RS-MDA-52-7443_IPF_AuxProductSpecification is broken<br>Update the link to Meteorological Bulletin M3.1. |
| | | Appendix A | [IPF-238] Update the AUX Product definition appendix<br>Update of s1-aux-pp2.xsd in with new parameter activateNoiseCorrection |
| | | Table 4-19 | [IPF-132] Clarified role of pixelSpacing parameter |
| 3/3 | April 4, 2017 | Table 7-9 | Increase cardinality gmfIndex (to support different gmf for VV and HH polarization) |
| | | Table 7-6 | Adding of two new inputs:<br>- activateTotalHs to trigger the activation of the calculation of the total significant wave height<br>- activateGroupDir to trigger the activation of the calculation of the swell direction from Group analysis |
| | Oct. 4, 2017 | Table 4-5, 4-9 | AUX_PP1 file updated to include two new fields: correctBistaticDelayMethod and estimateNoiseEquivalentPowerFlag |
| | | Table 6-11 | AUX_INS file updated to include a new field: azimuthTimeBias |
| | | Table 7-9 | Change on the PrIndex value description (to be conformed with the actual IPF behavior) |
| | | Table B-6 | Add a filed change Description in the manifest files |
| | 21/12/2017 | Table 12-1 | Correction of filenaming convention of the NetCDF file provided in the AUX_WAV product in order to be aligned with the real format (both before and after update applied on 2017-12-18) |
| 3/4 | 10/12/2019 | Section 8 – info table | Update for IPF 3.2:<br>- Ticket IPF-427: Introduce the AUX_PREORB |

| ISSUE | DATE | PAGE(S) | DESCRIPTION |
|---|---|---|---|
| | | | orbit types (Table 3-1 and section 9.1)<br>Update for IPF 3.3:<br>- Ticket IPF-422: Introduction of a new parameter ("MTFWV") in the simulated cross-spectra auxiliary files to adjust the effective Hs (section 8)<br>Update for IPF Future<br>- MPCS-2160: Introduction of optionalattribute in PP2 for element activateTotalHs to allow decoupling the activation depending on the beam (ie WV1 or WV2) (oswProcParams Table 7.6 Section 7.3.1) |
| 3/5 | 10/12/2020 | Section 11.1 | Clarification on the content of AUX_WND products |
| | | Section 12.1 | Clarification on the content of AUX_WAV products |
| | | Section 13.1 | Clarification on the content of AUX_ICE products |
| 3/6 | 25/02/2020 | Table 4-12<br>Table 6-5 | Added AUX_INS and AUX_PP1 paramters used for RFI detection and mitigation. |
| | | Table 7-9 | Clarification of the allowed values for gmf_index |
| 3/7 | 04/06/2021 | Table 4-12<br>Table 6-2 | Fixed the description of rfiMitigationDomain to reflect the processor's behaviour.<br>Added the deltaTXLatch parameter |
| 3/8 | 17/12/2021 | Table 7-6 | Update related to IPF 3.50<br>AUX_PP2: Adding new parameter for OSW processing parameters: activateNoiseCorrection ; velthresh |
| 3/9 | 15/02/2022 | Table 7-6 | Clarification<br>AUX-PP2: clarification of type and cardinality of the parameters and usage vs version of the processor |
| 3/10 | 17/11/2022 | Section 11 | Clarification on the spatial and temporal resolution of the ECMWF atmospheric model auxiliary data |

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AAEP | Azimuth Antenna Element Pattern |
| AAP | Azimuth Antenna Pattern |
| AN | Azimuth Notch |
| ANX | Ascending Node Crossing |
| ASAR | Advanced Synthetic Aperture Radar |
| AUX | Auxiliary |
| BAQ | Block Adaptive Quantization |
| CCN | Contract Change Notice |
| CDR | Critical Design Review |
| CDRL | Contract Data Requirements List |
| CLS | Collecte Localisation Satellites |
| dB | Decibel |
| DCE | Doppler Centroid Estimation |
| DEM | Digital Elevation Model |
| DSR | Data Set Record |
| EAP | Elevation Antenna Pattern |
| ECC | Event Control Code |
| ECMWF | European Centre for Medium-Range Weather Forecasts |
| EN | Elevation Notch |
| ESA | European Space Agency |
| ESRIN | European Space Research Institute |
| EW | Extra Wide Swath |
| FDBAQ | Flexible Dynamic Block Adaptive Quantization |
| GETASSE | Global Earth Topography and Sea Surface Elevation |
| GPS | Global Positioning System |
| GMF | Geophysical Model Function |
| GRD | Ground Range Detected |
| HH | Horizontal polarisation (Tx & Rx) |
| HV | Horizontal Vertical polarisation |
| Hz | Hertz |
| ICD | Interface Control Document |

| | |
|---|---|
| IPF | Instrument Processing Facility |
| IQ | In-phase, Quadrature |
| IRW | Impulse Response Width |
| ISLR | Integrated Side Lobe Ratio |
| ISP | Instrument Source Packet(s) |
| IW | Interferometric Wide Swath |
| KB | Kilo Byte |
| L1 | Level 1 |
| L2 | Level 2 |
| LUT | Lookup Table |
| MDA | MacDonald, Dettwiler and Associates Ltd. |
| MTF | Modulation Transfer Function |
| NaN | Not a Number |
| NRCS | Normalised Radar Cross Section |
| NRL | Normalised Reconstruction Level |
| NetCDF | Network Common Data Form |
| OCN | Ocean Product |
| OSW | Ocean Swell Spectra Component |
| OWI | Ocean Wind Field Component |
| PDR | Preliminary Design Review |
| PG | Processing Gain |
| POD | Precise Orbit Determination |
| PRF | Pulse Repetition Frequency |
| PRI | Pulse Repetition Interval |
| PSLR | Peak Side Lobe Ratio |
| RDB | Radar Database |
| RID | Review Item Disposition |
| RMS | Root Mean Squared |
| RVL | Radial Surface Velocity Component |
| SAR | Synthetic Aperture Radar |
| SAS | SAR Antenna Subsystem |
| SES | SAR Electronic Subsystem |
| SLC | Single Look Complex |
| SM | Stripmap |

| | |
|---|---|
| SOW | Statement Of Work |
| SPPDU | Space Packet Protocol Data Unit |
| SRGR | Slant Range to Ground Range |
| TBC | To Be Confirmed |
| TBD | To Be Determined |
| TOPSAR | Terrain Observation with Progressive Scanning SAR |
| URL | Uniform Resource Locator |
| UTC | Universal Time Coordinated |
| VH | Vertical Horizontal polarisation |
| VV | Vertical polarisation |
| W3C | World Wide Web Consortium |
| WV | Wave |
| XFDU | XML Formatted Data Unit |
| XML | eXtensible Markup Language |

# 1 INTRODUCTION

This section introduces the purpose, scope and structure of the document.

## 1.1 Purpose

This document describes the auxiliary data required by the Sentinel-1 Instrument Processing Facility (IPF) to perform L1 and L2 processing. It defines the content and format of auxiliary data files within the scope of this document (see Section 1.2) and provides references for the governing documentation of the auxiliary data files that are beyond the scope of this document and covered by other documents.

## 1.2 Scope

This specification satisfies the Sentinel-1 IPF input parameter files definition (deliverable IPAD-1) from the ESA Statement of Work (SOW) [A-1] with the modifications described in the Sentinel-1 IPF Contract Change Notice N. 2, Changes in ESRIN Contract No. 21722/08/I-LG [A-4].

This document specifies the content and format of the following auxiliary products and component files:

- L1 Processor parameters auxiliary data
- Calibration auxiliary data
- Instrument auxiliary data
- L2 Processor parameters auxiliary data
- Simulated Cross Spectra auxiliary data

There are other auxiliary data used by the Sentinel-1 IPF; however the definitions of the format and content of these data are beyond the scope of this document. The purposes of these auxiliary data and the way in which they must be structured for use by the IPF are described in the latter sections of this document and the reader is referred to the governing documentation for each.

The Sentinel-1 auxiliary data products defined by this document are a specialisation of the SAFE format following the SAFE recommendations for specialisation [A-3]. The implementation of the SAFE specialisation is managed in the XML schema files contained in Appendix A and is otherwise omitted from this specification as the details of the SAFE core specification are available in [A-2] and the details of the SAFE specialisations are available in [A-3].

## 1.3 Document Structure

This document is structured as follows:

- **Section 1** introduces the purpose, scope and structure of the document;
- **Section 2** lists the applicable and reference documents;
- **Section 3** presents an overview of the auxiliary data product format and defines the naming standard for auxiliary products and file components;
- **Section 4** describes the L1 processor parameters auxiliary data and presents an overview of its structure and content;
- **Section 5** describes the calibration auxiliary data and presents an overview of its structure and content;
- **Section 6** describes the instrument auxiliary data and presents an overview of its structure and content;
- **Section 7** describes the L2 processor parameters auxiliary data and presents an overview of its structure and content;
- **Section 8** describes the Simulated Cross Spectra auxiliary data and presents an overview of its structure and content;
- **Section 9** describes the orbit auxiliary data and provides a reference to its definition;
- **Section 10** describes the attitude auxiliary data and provides a reference to its definition;
- **Section 11** describes the ECMWF atmospheric model auxiliary data, presents its structure and provides a reference to its definition;
- **Section 12** describes the WAVEWATCH III model auxiliary data, presents its structure and provides a reference to its definition;
- **Section 13** describes the sea ice auxiliary data, presents its structure and provides a reference to its definition;
- **Section 14** describes the Excitation Coefficients Error matrix auxiliary data, presents its structure and provides a reference to its definition;
- **Appendix A** contains the XML schemas that define the content and format of the auxiliary data defined by this document. The schema files in Appendix A are the definitive source for content and format of each auxiliary data product described herein and are intended for an audience with an understanding of XML schema; and,
- **Appendix B** contains a high-level description of the SAFE manifest file and the mandatory components for the Sentinel-1 auxiliary data products.

# 2 DOCUMENTS

## 2.1 Applicable Documents

The following documents of the date/revision indicated form part of this document to the extent referenced herein.

| | | |
|---|---|---|
| A-1 | GMES-DFPR-EOPG-SW-07-00006 | Sentinel-1 Product Definitions & Instrument Processing Facility Development Statement of Work, Issue/Revision 4/1, 23-05-2008. ESA. |
| A-2 | PGSI-GSEG-EOPG-FS-05-0001 | Standard Archive Format for Europe (SAFE) Control Book Volume 1 Core Specifications, Issue 1/8, Jun. 28, 2009. ESA. |
| A-3 | PGSI-GSEG-EOPG-FS-05-0002 | Standard Archive Format for Europe (SAFE) Control Book Volume 2 Recommendation for specialisations Issue/Revision 1/7. Jun. 28, 2009. ESA. |
| A-4 | CCN No. 2 | Contract Change Notice N. 2, Changes in ESRIN Contract No. 21722/08/I-LG, June 21, 2010. MacDonald Dettwiler. |
| A-1 | 01-7416A | MDA Proposal to ESA for Sentinel-1 IPF Contract Change Request #04, Nov. 13, 2012. |
| A-2 | GMES-GSEG-EOPG-SW-12-0037 | Sentinel-1 IPF Development Change Request No. 4, Issue/Revision 1/2, Oct. 25, 2012. ESA. |
| A-3 | GMES-GSEG-EOPG-FS-10-0075 | Sentinels POD File Format Specification. Issue/Revision 2/3, Sep 16, 2019. ESA. |
| A-4 | EO-MA-DMS-GS-0007 | Earth Observation Mission CFI Software EO_DATA_HANDLING Software User Manual Issue/Revision 4/2. July 31, 2011. ESA. |
| A-5 | S1-IC-MDA-52-7454 | Sentinel-1 IPF Interface Control Document. Issue/Revision 1/7. Feb. 18, 2013. MacDonald Dettwiler. |

| | | |
|---|---|---|
| A-6 | | European Centre for Medium-Range Weather Forecasts (ECMWF) Deterministic Atmospheric Model Products in GRIB format. Meteorological Bulletin M3.1. http://www.ecmwf.int/sites/default/files/3.1.pdf, section 4.2 Products in GRIB code |
| A-7 | | http://www.gebco.net/data_and_products/gridded_bathymetry_data/ WAVEWATCH III Model. IFREMER. ftp://ftp.ifremer.fr/ifremer/cersat/products/gridded/wavewatch3/HINDCAST/README |
| A-8 | | Sea Ice Product Manual. Version 3.6. September 2009. OSI SAF. http://www.osi-saf.org/biblio/docs/ss2_pmseaice_3_6.pdf |
| A-9 | ST-ESA-S1QC-PFS-002 | Sentinel-1 Quality Control [SD-05-1300-2] QCSS Error and Failure Matrices Format Specification. Issue/Revision 1/1 Nov. 15, 2011. S&T. |
| A-10 | S1-IF-ASD-PL-0007 | Sentinel-1 SAR Space Packet Protocol Data Unit, Issue 8, Aug. 23, 2012. EADS Astrium. |
| A-11 | S1-TN-MDA-52-7445 | Sentinel-1 Level 1 Detailed Algorithm Definition. Issue/Revision 1/4. Sep. 27, 2012. MDA. |

## 2.2 Reference Documents

The following documents provide useful reference information associated with this document. These documents are to be used for information only and changes to the date/revision number (if provided) shall not make this document out of date.

| | |
|---|---|
| R-1 | XML 1.1 (Second Edition), W3C Recommendation, 16 August 2006, Bray et al. |
| R-2 | XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004, Thompson et al. |

R-3 XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004, Biron et al.

R-4 S1-TN-ASD-PL-0018 Sentinel-1 SAR Instrument Definition of Roll Steering Law, Issue 1, July 18, 2008. EADS Astrium.

# 3 AUXILIARY DATA OVERVIEW

Auxiliary data is information required for the processing of L1 and L2 Synthetic Aperture Radar (SAR) products that is not available in the Instrument Source Packets (ISP) and must be retrieved from external inputs. This auxiliary data is used by the Sentinel-1 IPF during processing as summarised in Table 3-1 below.

**Table 3-1 Summary of Auxiliary Data used by Sentinel-1 IPF**

| Auxiliary Data | Description | Proc. Level | Product ID | Reference |
|---|---|---|---|---|
| L1 Processor Parameters | The Processor Parameters Auxiliary Data contains parameters and options for configuring the L1 processing performed within the IPF. | L1 | AUX_PP1 | Section 4 |
| Calibration | The Calibration Auxiliary Data contains parameters used to correct and calibrate the imagery during processing. | L1 | AUX_CAL | Section 5 |
| Instrument | The Instrument Auxiliary Data contains information about the SAR instrument that is required by the IPF for image processing. | L1 | AUX_INS | Section 6 |
| L2 Processor Parameters | The Processor Parameters Auxiliary Data contains parameters and options for configuring the L2 processing performed within the IPF. | L2 | AUX_PP2 | Section 7 |
| Simulated Cross Spectra | The OSW processing algorithm requires a look-up table to predict the modulation transfer function (MTF) and to remove non-linear effects from the cross spectra. | L2 | AUX_SCS | Section 8 |
| Orbit | Orbit auxiliary data contain information about the position of the satellite during the acquisition of SAR data. There are three types of orbit auxiliary data:<br>• Precise Orbit Ephemeris;<br>• Restituted Orbit; and<br>• Predicted Orbit | L1 | AUX_POE<br>AUX_RES<br>AUX_PRE | Section 9 |
| Attitude | Attitude auxiliary data contain information about the pointing of the satellite during the acquisition of SAR data. | L1 | AUX_ATT | Section 10 |
| ECMWF Atmospheric Model | The ECMWF atmospheric model provides wind speed and direction at 10 m above the sea surface. | L2 | AUX_WND | Section 11 |
| WAVEWATCH III Model | The WavewatchIII model provides surface stockes drift velocity and direction necessary for the interpretation of the RVL component of the L2 OCN product. | L2 | AUX_WAV | Section 12 |

| Auxiliary Data | Description | Proc. Level | Product ID | Reference |
|---|---|---|---|---|
| Sea Ice Data | Sea ice auxiliary data is used to estimate the percentage of the image covered by ice. Used to process the OWI component of the L2 OCN products. | L2 | AUX_ICE | Section 13 |
| Excitation Coefficients Error Matrix | Excitation coefficients error matrix auxiliary data is used to derive accurate Doppler estimation using an antenna model for the RVL component of the L2 OCN product. | L2 | AUX_ECE | Section 14 |

## 3.1 Format Overview

Most Sentinel-1 auxiliary data products are a collection of files that are grouped together and formatted according to the SAFE core specification [A-2]. Note that Orbit Data (Section 8), Attitude Data (Section 10) and Excitation Coefficients Error Matrix Data (Section 14) are exceptions, as these are not wrapped in SAFE format, and their formats are described in their respective sections. An auxiliary data product that is SAFE formatted is a file folder which contains the following auxiliary data file components:

- Manifest file;
- Auxiliary data file(s); and,
- Representation data file(s).

The manifest file is an eXtensible Markup Language (XML) file [R-1] formatted according to [A-2] that serves two important purposes within the product: it contains information about the data files that make up the auxiliary product, and it contains general information that describes properties of the auxiliary product.

The auxiliary data file is an XML file that contains the specific information required by the Sentinel-1 IPF for image processing. The exact content and format of the auxiliary data file depends on the type of information that it contains.

Representation data files are XML Schema files composed of XML schema structures [R-2] and data types [R-3] that define the format and content of the auxiliary data file to which they apply. The representation data file can be used by XML tools to validate the structure and content of the auxiliary file and it can be used by humans and parsing tools to read, to manipulate and to exploit the values of the data contained in the relevant auxiliary data file. The representation data files form the definitive source for the content and format of each of the auxiliary data files described in this document.

Figure 3-1 presents a graphical overview of the logical structure of Sentinel-1 auxiliary data products and Figure 3-2 presents a graphical overview of the physical structure of Sentinel-1 auxiliary data products.

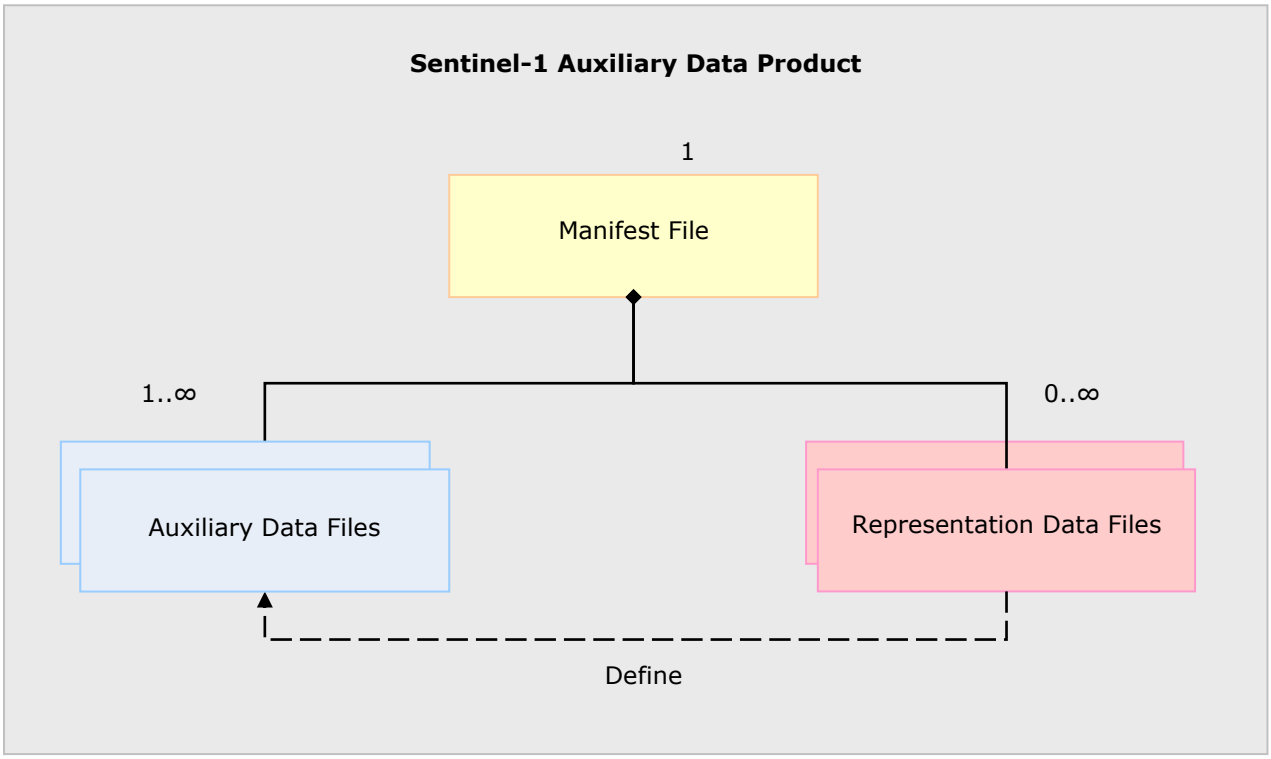


**Figure 3-1 Sentinel-1 Auxiliary Data Product Logical Structure**

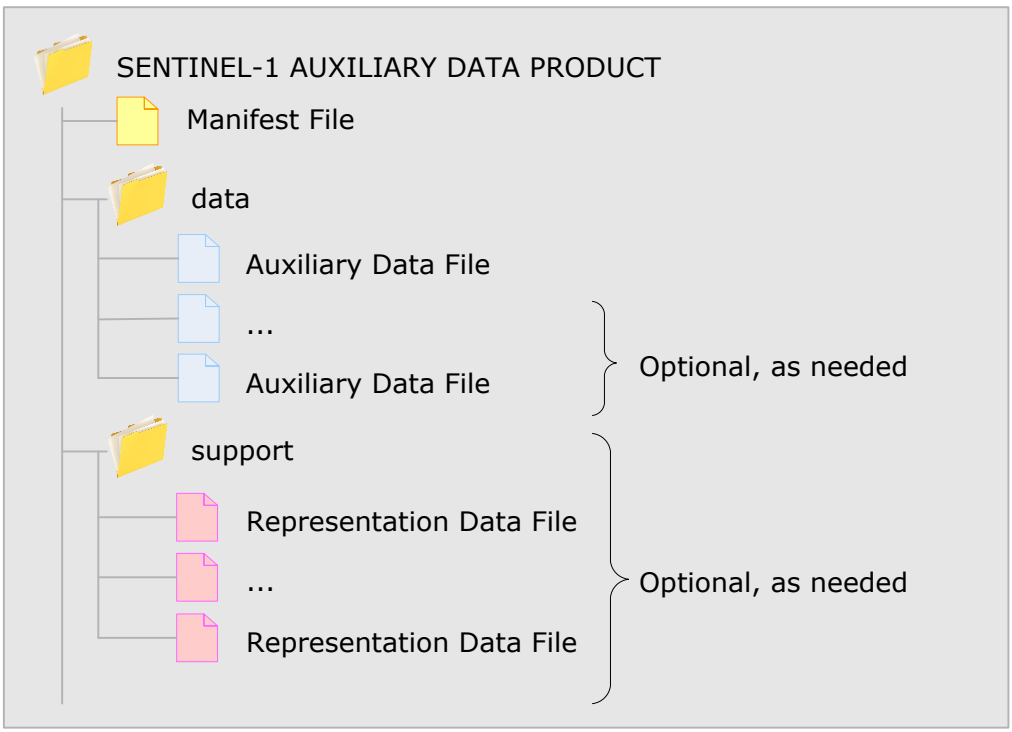


**Figure 3-2 Sentinel-1 Auxiliary Data Product Physical Structure**

## 3.2 Naming Standard

This section defines the naming standard for the auxiliary data products and the component files within the product. This standard is used for most of the auxiliary data products described by this document, which are all wrapped in SAFE formatting. Auxiliary data files that are not SAFE formatted, such as Orbit Data (Section 8), Attitude Data (Section 10) and Excitation Coefficients Error Matrix Data (Section 14) use their own naming conventions, as specified in their respective sections.

### 3.2.1 Product

The naming standard for Sentinel-1 auxiliary data products contains uppercase alphanumeric characters separated by underscores "_" as illustrated in Figure 3-3. Table 3-2 presents each naming element used and the description and range of each.

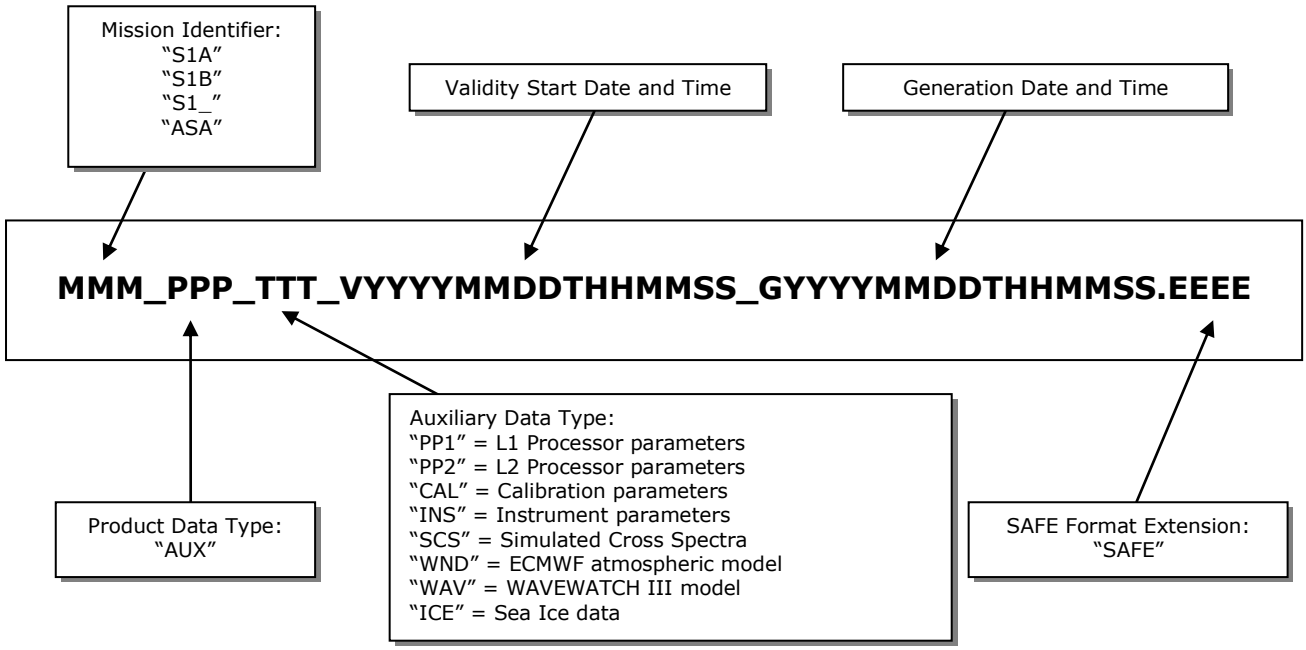


**Figure 3-3 Auxiliary Product Naming Standard**

**Table 3-2 Sentinel-1 Auxiliary Product Naming Elements**

| Name Element Pattern | Name Element Description | Name Element Range |
|---|---|---|
| MMM | Mission Identifier | S1A, S1B, S1_, ASA |
| PPP | Product data type | AUX |
| TTT | Auxiliary file data type | PP1, PP2, CAL, INS, SCS, WND, WAV, ICE |

| Name Element Pattern | Name Element Description | Name Element Range |
|---|---|---|
| VYYYYMMDDTHHMMSS | Date and time when the product becomes valid. | V followed by fourteen digits representing the date and time separated by the character T. |
| GYYYYMMDDTHHMMSS | Date and time when the product was generated. | G followed by fourteen digits representing the date and time separated by the character T. |
| EEEE | Product format extension | SAFE |

Within the auxiliary product name, the mission identifier is always set to “S1A” or “S1B” to identify the auxiliary product as Sentinel-1A or Sentinel-1B data respectively or set to “S1_” for auxiliary data that is not satellite-specific. The mission identifier “ASA” identifies auxiliary products required for ASAR data processing within the Stand-Alone S1 IPF.

The product data type is always set to “AUX” to identify the product as an auxiliary data product. The auxiliary file data type varies with the information contained in the product and is set accordingly.

The validity start date and time defines when the auxiliary data becomes applicable and the file generation data and time defines when the auxiliary data was created. The criteria for selecting an appropriate auxiliary product to use when processing a particular job order are as follows:

1. Select the auxiliary product(s) with a validity start date/time closest to, but not later than, the start of the job order;
2. If there is more than one product which meets the first criteria (e.g. two auxiliary files have the same validity date/time), then use the auxiliary product with the latest generation time.

### 3.2.2 Data Sets

The naming standard for Sentinel-1 auxiliary data sets is composed of lowercase alphanumeric characters separated by hyphens “-”. Table 3-3 defines the naming elements and their range and is used through the remainder of this document as the standard for naming Sentinel-1 auxiliary data sets.

**Table 3-3 Sentinel-1 Auxiliary Product Naming Elements**

| Name Element Pattern | Name Element Description | Name Element Range |
|---|---|---|
| mmm | Mission Identifier | s1a, s1b, s1-, asa |
| aaa | File data type | aux |
| ttt | Auxiliary file data type | pp1, pp2, cal, ins, scs |
| sss | Swath identifier (up to 3 lower case alphanumeric characters). | s1, s2, s3, s4, s5, s6, iw, iw1, iw2, iw3, ew, ew1, ew2, ew3, ew4, ew5, wv1, wv2, en, n1, n2, n3, n4, n5, n6, is1, is2, is3, is4, is5, is6, is7 |

| Name Element Pattern | Name Element Description | Name Element Range |
|---|---|---|
| pp | Polarisation | hh (Single HH polarisation), hv (Single HV polarisation), vv (Single VV polarisation), vh (Single VH polarisation) |
| eeee | File format extension | xml, info, nlin, grb, nc |

Within the auxiliary data file name, the mission identifier is always set to “s1a” or “s1b” to identify the auxiliary file as Sentinel-1A data or Sentinel-1B data respectively or set to “s1-” for auxiliary data that is not satellite-specific. The mission identifier “asa” identifies auxiliary files required for ASAR data processing within the Stand-Alone S1 IPF.

The product data type is always set to “aux” to identify the file as an auxiliary data file. The auxiliary file data type varies with the information contained in the file and is set accordingly.

Files within the auxiliary product exempt from this naming standard are:

- Any data files defined by an external specification;
- Any external representation data files required to support the auxiliary product (the s1a-object-types.xsd file for example); and,
- The manifest file which is always named “manifest.safe” in accordance with [A-2].

## 3.3 Content Table Conventions

Sections 4.3, 5.3, 6.3, 7.3 and 8.3 present tables detailing the content of the auxiliary data defined by this specification. In order to help understand the information contained in these tables the columns present in each table and their meaning are presented in Table 3-4.

**Table 3-4 Auxiliary Data Content Table Column Descriptions**

| Column Name | Description |
|---|---|
| Name | Defines the name of the element or attribute.<br>Elements are displayed in normal font.<br>*Attributes are displayed in italics.* |
| Description | Describes the purpose of the element or attribute including the units (if applicable). The absence of a unit designation for string data types implies a string literal and for numerical data types (integers, floats, etc…) the absence of a unit designation implies an absolute number.<br>Elements are displayed in normal font.<br>*Attributes are displayed in italics.* |

| Column Name | Description |
|---|---|
| Data Type | Defines the data type of the element or attribute.<br>Elements are displayed in normal font.<br>*Attributes are displayed in italics.* |
| Cardinality | Elements are displayed in normal font and this column defines the number of occurrences of the element in the form:<br>[minOccurs ..] maxOccurs<br>*Attributes are displayed in italics and this column describes the use of the attribute where, "optional" means the attribute may or may not be present and "required" means the presence of the attribute is mandatory.* |

## 3.4 Primitive Data Types

Table 3-5 describes the primitive data types defined in the Sentinel-1 namespace.

Note that ASAR options are only used for ASAR offline testing and will not be used operationally.

**Table 3-5 Primitive Data Types used in Sentinel-1 Auxiliary Data**

| Type Name | Description | Range |
|---|---|---|
| chirpSourceType | Enumeration of the available chirp schemes. | Extracted, Nominal |
| pgSourceType | Enumeration of the available PG schemes. | Extracted, Model |
| rrfSpectrumType | The type of range matched filter to use during processing. "Unextended": range reference function is unextended in frequency domain; "Extended Flat": range reference function is extended and flat in frequency domain; and, "Extended Tapered": range reference function is extended and tapered in frequency domain. | Unextended, Extended Flat Extended Tapered |
| dcMethodType | Enumeration of Doppler centroid calculation/estimation methods. | Data Analysis, Orbit and Attitude, Pre-defined |
| dcInputDataType | Enumeration of input data types for Doppler centroid estimation. | Raw, Range Compressed |

| Type Name | Description | Range |
|---|---|---|
| swathType | Enumeration of all valid swath identifiers for the Sentinel-1 SAR instrument. The S1-S6 swaths apply to SM products, the IW and IW1-3 swaths apply to IW products (IW is used for detected IW products where the 3 swaths are merged into one image), the EW and EW1-5 swaths apply to EW products (EW is used for detected EW products where the 5 swaths are merged into one image), and the WV1-2 swaths apply to WV products. The EN, N1-N6 swaths apply to the Sentinel-1 notch modes used for instrument calibration. The RF swath applies to the Sentinel-1 RFC mode which is not processed by the IPF. The IS1-IS7 swaths apply to ASAR IM and WV products. | S1, S2, S3, S4, S5, S6, IW, IW1, IW2, IW3, EW, EW1, EW2, EW3, EW4, EW5, WV, WV1, WV2, EN, N1, N2, N3, N4, N5, N6, RF, IS1, IS2, IS3, IS4, IS5, IS6, IS7 |
| swathNumberType | Swath number from SPPDU [A-10]. | 0 .. 127 |
| weightingWindowType | Enumeration of weighting window names. | Hamming, None |
| polarisationType | Enumeration of valid polarisations for the Sentinel-1 SAR instrument. | HH, HV, VH, VV |
| rxPolarisationType | Enumeration of valid receive polarisations for the Sentinel-1 SAR instrument. | H, V |
| sensorModeType | Enumeration of the SAR instrument imaging modes. | S1, S2, S3, S4, S5, S6, IW, EW, WV, EN, N1, N2, N3, N4, N5, N6, RF, IM |
| signalType | Enumeration of valid signal types for the Sentinel-1 SAR instrument. | Echo, Noise, TxCal, RxCal, EpdnCal, EpdnCalIso, TaCal, ApdnCal, TaRxCal, ApdnRxCal, TxRxOff, Silent |
| bandwidthType | Enumeration of the available bandwidths for SAR signals in the timeline. | Image, Full |
| outputPixelsType | Enumeration of output pixel data types. | 32 bit Float, 16 bit Signed Integer, 16 bit Unsigned Integer, 8 bit Unsigned Integer |
| topsFilterConventionType | Enumeration of valid conventions for defining the TOPS ramping/de-ramping filter. | All Lines, Only Echo Lines |
| calCombinationMethodType | Enumeration of valid methods for combining calibration pulses. | PCC2, Average, Isolation Subtraction |
| bool | Boolean (true or false). | true, false |
| string | Character string. | 1 .. 512 UTF-8 characters. |
| unsignedInt | 32 bit unsigned integer. | 0 .. 4294967295 |
| int16 | 16 bit signed integer. | -32768 .. 32767 |
| int32 | 32 bit signed integer. | -2147483648 .. 2147483647 |
| uint32 | 32 bit unsigned integer. | 0 .. 4294967295 |

| Type Name | Description | Range |
|---|---|---|
| byte | 8 bit signed byte. | -128 .. 127 |
| ubyte | 8 bit unsigned integer. | 0 .. 255 |
| uint16 | 16 bit unsigned integer. | 0 .. 65535 |
| uint32 | 32 bit unsigned integer. | 0 .. 4294967295 |
| float | 32 bit single precision floating point number with an optional "units" attribute. | Machine dependent |
| double | 64 bit double precision floating point number with an optional "units" attribute. | Machine dependent |
| complex | 64 bit complex number consisting of a 32 bit single precision floating point real part and a 32 bit single precision floating point imaginary part. | Machine dependent |
| floatArray | String containing an array of single precision floating point values separated by spaces. The mandatory count attribute defines the number of elements in the array. | 0 .. 65535 float values |
| doubleArray | String containing an array of double precision floating point values separated by spaces. The mandatory count attribute defines the number of elements in the array. | 0 .. 65535 double values |
| complexArray | String containing an array of complex single precision floating point values separated by spaces. The mandatory count attribute defines the number of elements in the array. | 0 .. 65535 complex values |
| intArray | String containing an array of int values separated by spaces. The mandatory count attribute defines the number of elements in the array. | 0 .. 65535 integer values |
| floatCoefficientArray | String containing an array of float values separated by spaces. This array is sized for a field which represents a set of coefficients. The mandatory count attribute defines the number of elements in the array. | 0 .. 22 float values |
| doubleCoefficientArray | String containing an array of double values separated by spaces. This array is sized for a field which represents a set of coefficients. The mandatory count attribute defines the number of elements in the array. | 0 .. 22 double values |
| floatPatternArray | String containing an array of float values separated by spaces. This array is sized for a field which represents an antenna pattern. The mandatory count attribute defines the number of elements in the array. The Sentinel-1 IPF supports antenna patterns of up to 4500 values. | 0 .. 4500 float values |

# 4 L1 PROCESSOR PARAMETERS AUXILIARY DATA

## 4.1 Overview

The L1 processor parameters auxiliary file contains parameters required by the IPF for producing L1 products with the desired properties. It allows the fine-tuning of the processor and the selection of specific processing options.

This auxiliary product is expected to be updated infrequently and will change only when the characteristics of the output products need to be modified or when the IPF software is modified.

### 4.1.1 Product Structure

Table 4-1 describes the physical structure and data file components included in the L1 Processor Parameters Auxiliary Product as well as a nominal estimated size for each.

**Table 4-1 Sentinel-1 L1 Processor Parameters Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| MMM_AUX_PP1_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| mmm-aux-pp1.xml | Auxiliary Data | 1 | 370 KB |
| support/ | Directory | 1 | N/A |
| s1-aux-pp1.xsd | Representation Data | 1 | 46 KB |
| s1-object-types.xsd | Representation Data | 1 | 48 KB |

### 4.1.2 Naming Standard

Naming for the processor parameters auxiliary product follows the standard defined in Section 3.2.1 and naming of the component files within the processor parameters auxiliary product follows the standard defined in Section 3.2.2.

## 4.2 Data File Definition

The schema file for the definition of the processor parameters auxiliary file is specified in Appendix A1.

## 4.3 Data File Content

Figure 4-1 presents a high-level graphical view of the processor parameters auxiliary data contained in the auxiliary data file mmm-aux-pp1.xml, and a detailed tabular view of the processor parameters auxiliary file is presented in the tables below.

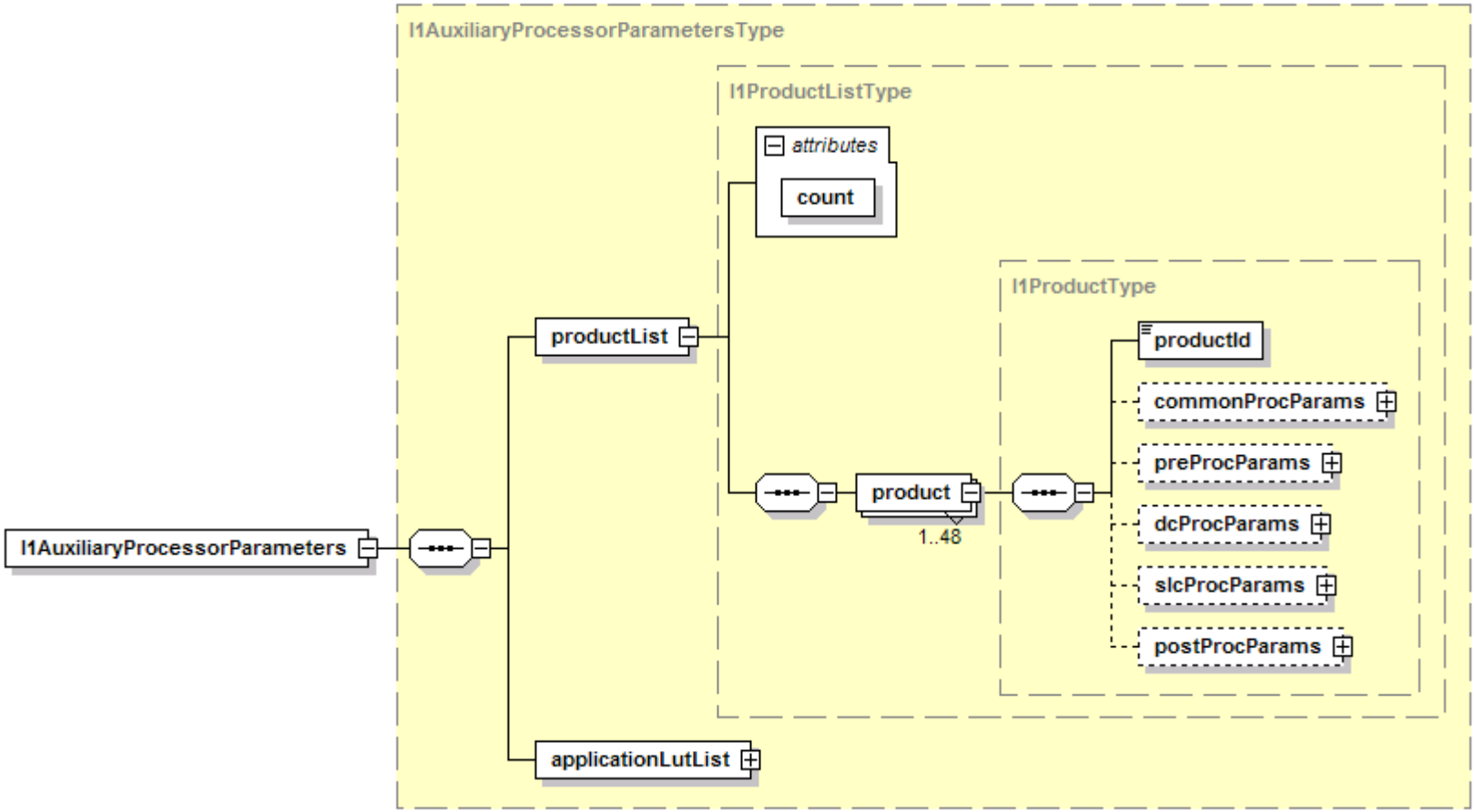


**Figure 4-1 L1 Auxiliary Processor Parameters**

**Table 4-2 Element – l1AuxiliaryProcessorParameters**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| productList | List of L1 products containing the applicable auxiliary parameters for each. This list contains an entry for each product the IPF is capable of generating, indexed by its unique product identifier. | l1ProductListType | 1 |
| applicationLutList | List of application LUTs. This element is a list of all available application LUTs. The application identifier used to index the list comes from the Job Order. | applicationLutListType | 1 |

**Table 4-3 Data Type - productListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *The number of elements contained in the list. There is an entry for each product type that the IPF is capable of generating.* | *unsignedInt* | *required* |
| product | Product auxiliary parameters. This DSR contains all of the auxiliary parameters required to process a single product. The parameters are stored in structures that are grouped together by the logical processing steps used during image creation. The parameters within this DSR and its children are not polarisation dependent and in general apply to the entire product; however, in some cases a distinction must be made amongst swaths and when this is necessary the records are indexed with a swath identifier. There are 16 standard product types defined in the ICD [A-5] for the IPF so in general there will be 16 product type entries in the list, however more products can be defined and therefore more than 16 product types are allowed in this list. | l1ProductType | 1 .. 48 |

**Table 4-4 Data Type - productType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| productId | Product type identifier to which this set of parameters applies. The productId is used to index and find the correct set of auxiliary parameters for each product the IPF is capable of generating. This field corresponds to the first 9 characters of the product type identifiers listed in the Job Order File_Type field. For example, the S1 IPF ICD [A-5] defines a product identifier for SM SLC standard products as "SM_SLC__1S", so the parameters that correspond to this product are identified by the string "SM_SLC__1". | string | 1 |
| commonProcParams | Common processing auxiliary parameters. This record holds the parameters that are common among multiple steps in the image processing chain. | commonProcParamsType | 0 .. 1 |
| preProcParams | Pre-processing auxiliary parameters. This record contains the auxiliary parameters required during image pre-processing. | preProcParamsType | 0 .. 1 |
| dcProcParams | Doppler centroid processing auxiliary parameters. This record contains the auxiliary parameters required during Doppler centroid processing. | dcProcParamsType | 0 .. 1 |
| slcProcParams | SLC processing auxiliary parameters. This record contains the auxiliary parameters required during SLC image processing. | slcProcParamsType | 0 .. 1 |
| postProcParams | Post processing auxiliary parameters. This record contains the auxiliary parameters required during image post processing. This includes: SLC post-processing, GRD processing and Quick-look image processing. | postProcParamsType | 0 .. 1 |

### 4.3.1 Common Parameters

The common processing parameters section contains the parameters that are common to or apply to more than one of the image processing steps. Figure 4-2 presents a high-level graphical overview of the common processing parameters and the subsequent tables describe the content of this DSR.

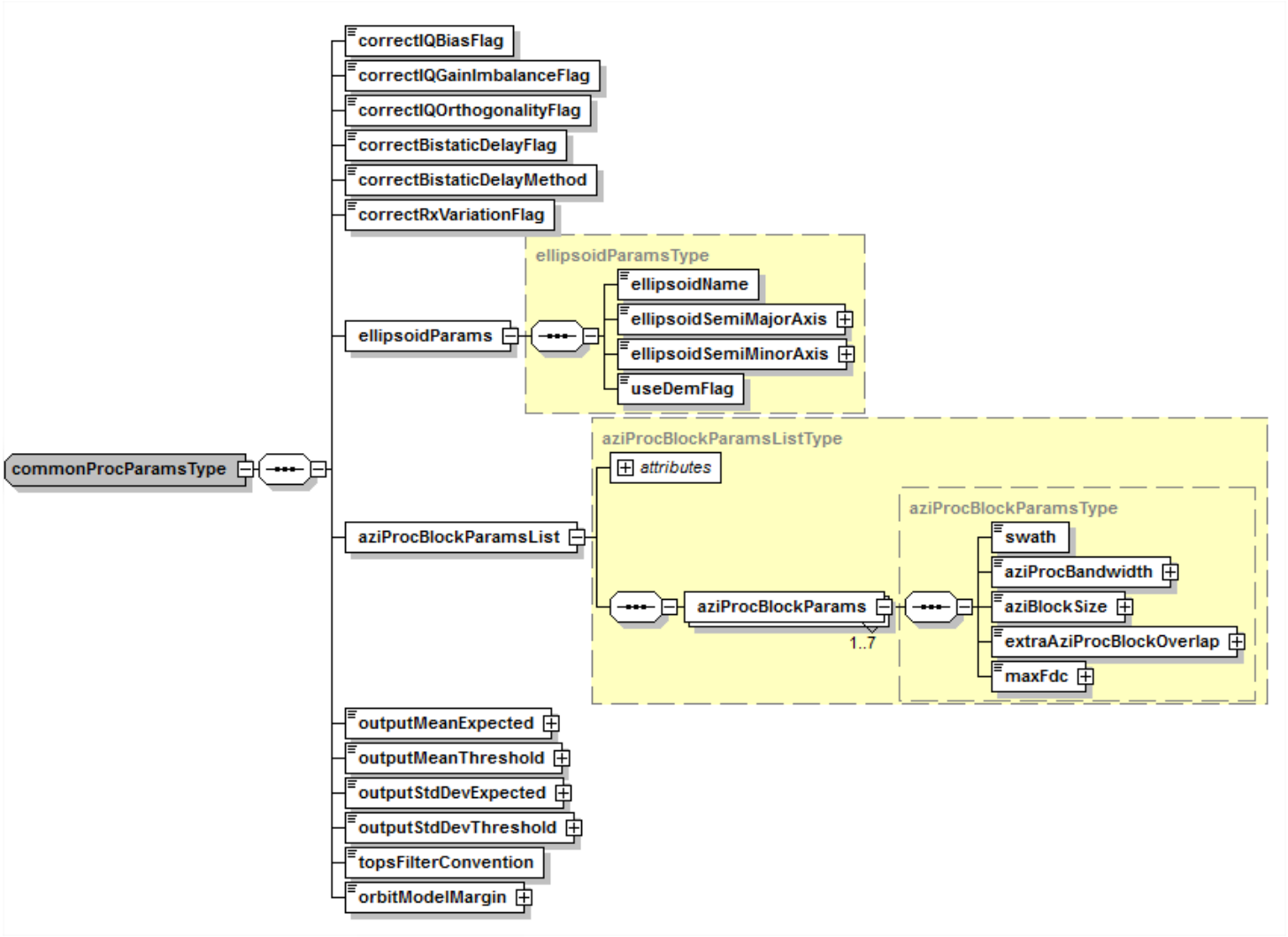


**Figure 4-2 Common Parameters Auxiliary Data**

**Table 4-5 Data Type - commonProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| correctIQBiasFlag | Flag to control the correction of the constant biases from the I and Q channels. IQ bias correction will be performed if and only if this flag is set to "true". | bool | 1 |
| correctIQGainImbalanceFlag | Flag to control the correction of the IQ gain imbalance. IQ gain imbalance correction will be performed if and only if this flag is set to "true". | bool | 1 |
| correctIQOrthogonalityFlag | Flag to control the correction of the IQ orthogonality. IQ orthorgonality correction will be performed if and only if this flag is set to "true". | bool | 1 |
| correctBistaticDelayFlag | Flag to compensate for the bi-static delay. Correction will be performed if and only if this flag is set to "true". | bool | 1 |
| correctBistaticDelayMethod | Method used to compensate for the bi-static delay. If set to "Coarse" then only the coarse bistatic delay correction is applied; otherwise, if set to "Fine" then the full correction is applied.<br>Flag introduced starting from IPF V290. | bistaticDelayMethodType | 1 |
| correctRxVariationFlag | Flag to control the correction of the gain variation across the receive window. Receive variation correction will be performed if and only this flag is set to “true”. | bool | 1 |
| ellipsoidParams | Ellipsoid and DEM parameters. | ellipsoidParamsType | 1 |
| aziProcBlockParamsList | Azimuth processing block parameters used for SM and IM processing. | aziProcBlockParamsListType | 1 |
| outputMeanExpected | This parameter specifies the expected mean of the samples in the output image and is used for verifying that the calculated mean of the output samples is within the tolerated threshold. | double | 1 |
| outputMeanThreshold | Threshold for setting the outputDataMeanOutsideNominalRange flag in the output product annotations. This is the value T, such that the measured mean must fall between the outputMeanExpected-T and outputMeanExpected+T. | double | 1 |
| outputStdDevExpected | This parameter specifies the expected standard deviation of the samples in the output image and is used for verifying that the calculated std. dev. of the output samples is within the tolerated threshold. | double | 1 |
| outputStdDevThreshold | Threshold for setting the outputDataStdDevOutsideNominalRange flag in the output product annotations. This is the value D, such that the measured standard deviation must fall between the | double | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| | outputStdDevExpected-D and outputStdDevExpected+D. | | |
| topsFilterConvention | Name of the TOPS filter convention to use during processing. This field controls how the TOPS ramping/de-ramping filters are defined. If set to "Only Echo Lines" then the filter is defined using only the echo lines in a burst; otherwise, if set to "All Lines" then the filter is defined using all the lines in a burst. | topsFilterOriginType | 1 |
| orbitModelMargin | Additional time to add to the start and end of the orbit model generated by the IPF [s]. This provides margin for performing interpolation near the boundaries of the sensing start and stop times and extrapolation beyond the boundaries of the sensing start and stop times.<br>For example, if the sensing start time is $T_{start}$, the sensing stop time is $T_{stop}$ and <orbitModelMargin>2.0</orbitModelMargin>, then the orbit model generated by the IPF will range from ($T_{start}$ - 2.0) .. ($T_{stop}$ + 2.0) | double | 1 |

**Table 4-6 Data Type - ellipsoidParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| ellipsoidName | Name of the reference ellipsoid. | string | 1 |
| ellipsoidSemiMajorAxis | Semi-major axis of ellipsoid [m]. | double | 1 |
| ellipsoidSemiMinorAxis | Semi-minor axis of ellipsoid [m]. | double | 1 |
| useDemFlag | This flag is used to control the use of a DEM during processing. Set to true if a DEM is to be used during processing, false otherwise. | bool | 1 |

**Table 4-7 Data Type - aziProcBlockParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of parameter records in this list. There is one record per swath* | *unsignedInt* | *required* |
| aziProcBlockParams | Azimuth processing block parameters record indexed by swath. There will be one record per swath. | aziProcBlockParamsType | 1 ... 7 |

**Table 4-8 Data Type - aziProcBlockParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Swath identifier. This parameter defines the swath to which this record applies. | swathType | 1 |
| aziProcBandwidth | Total processing bandwidth [Hz]. This parameter defines the bandwidth (Bw) to use during SM and WV SLC processing. The processing bandwidth (Bw) must be 0 < Bw <= PRF. This is used in the generation of the internal SL1 and SL2 products. | float | 1 |
| aziBlockSize | Azimuth processing block size [lines]. | uint32 | 1 |
| extraAziProcBlockOverlap | Extra azimuth block overlap to account for possible variation of Doppler centroid frequency from azimuth block to azimuth block [lines]. | uint32 | 1 |
| maxFdc | For SM this is the maximum expected absolute value of Doppler centroid frequency [Hz]. Used to calculate the SM SLC azimuth matched filter throwaway component of azimuth block overlap as applicable to all azimuth blocks in a segment. Although this field is an array, for SM only the first coefficient is applicable.<br>For TOPS this is the polynomial that describes the expected offset and maximum excursion of the Doppler centroid frequency over the total slant range time extent, $T_{SR}$, of all swaths in the mode. It is used to calculate the maximum burst size that will be used for the whole segmentThis is an array of five floating point coefficients separated by spaces. The first coefficient is the Doppler centroid offset [Hz] and the remaining coefficients describe the expected variation of the Doppler centroid frequency along range. The polynomial is evaluated as a function of the slant range time $t$ as:<br>$maxFdc(t) = C0 + C1 * t + C2 * t\text{^}2 + C3 * t\text{^}3 + C4 * t\text{^}4 \quad \{t = 0 .. T_{SR}\}$ | floatCoefficientArray | 1 |

### 4.3.2 Pre-Processing Parameters

The Pre-Processing Parameters section contains the parameters that control and fine tune the pre-processing of the IPF. Figure 4-3 presents a high-level graphical overview of the pre-processing parameters and the subsequent tables describe the content of this DSR.

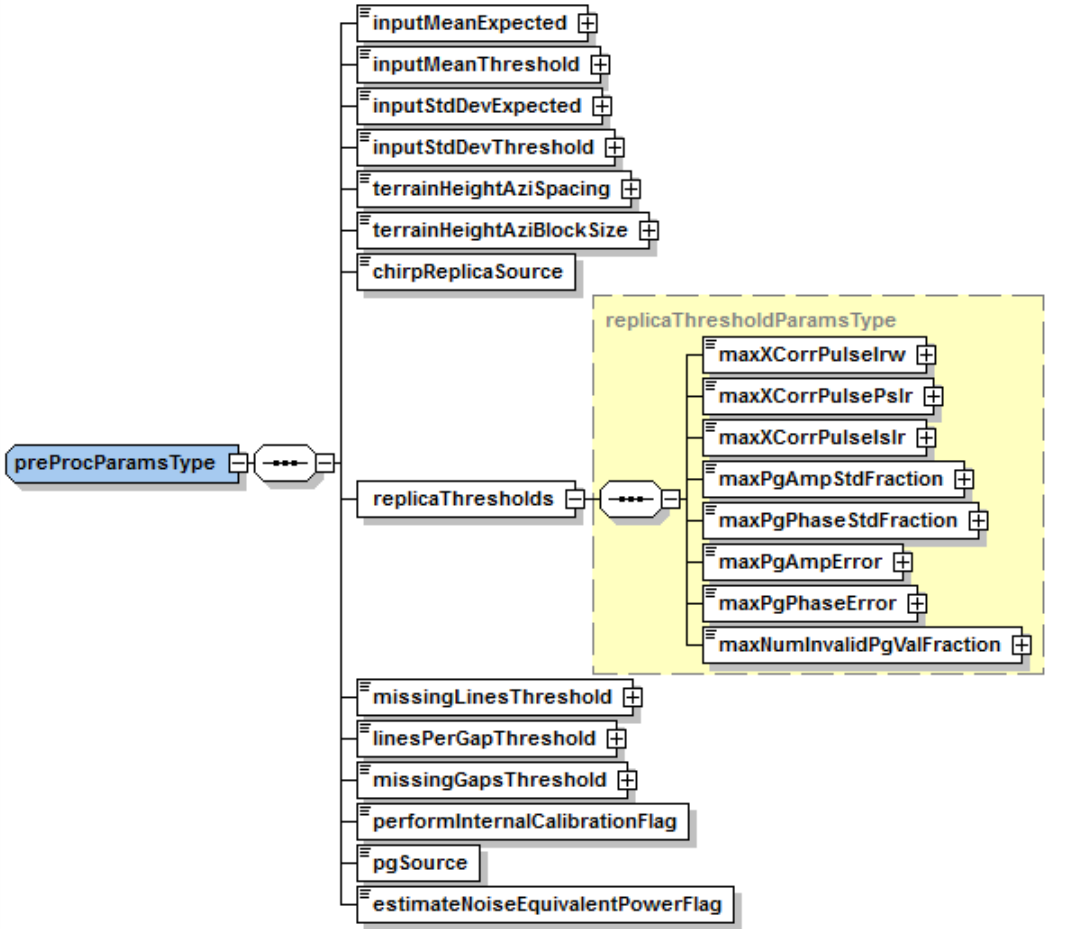


**Figure 4-3 Pre-processing Parameters Auxiliary Data**

**Table 4-9 Data Type - preProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| inputMeanExpected | This parameter specifies the expected mean of the input I and Q samples and is used in verifying that the calculated mean of the input samples is within the tolerated threshold. | double | 1 |
| inputMeanThreshold | Threshold for setting the inputDataMeanOutsideNominalRange flag in the output product annotations. This is the value T, such that the measured mean must fall between the inputMeanExpected-T and inputMeanExpected+T. This threshold is used for both the I and Q channels. | double | 1 |
| inputStdDevExpected | This parameter specifies the expected standard deviation of the input I and Q samples and is used in verifying that the calculated std. dev. of the input samples is within the tolerated threshold. | double | 1 |
| inputStdDevThreshold | Threshold for setting the inputDataStdDevOutsideNominalRange flag in the output product annotations. This is the value D, such that the measured std. dev. must fall between the inputStdDevExpected-D and inputStdDevExpected+D. This threshold is used for both the I and Q channels. | double | 1 |
| terrainHeightAziSpacing | Spacing of terrain height estimates along azimuth [s]. The calculation, terrainHeightAziSpacing + terrainHeightAziBlockSize, defines the location (relative to the start of the segment) and size of the azimuth blocks used to generate each terrain height estimate. These two parameters should be set to ensure that there is an overlapping region within each adjacent azimuth block. | double | 1 |
| terrainHeightAziBlockSize | Size of the block along azimuth used to calculate an average terrain height [s]. The calculation, terrainHeightAziSpacing + terrainHeightAziBlockSize, defines the location (relative to the start of the segment) and size of the azimuth blocks used to generate each terrain height. . These parameters should be set to ensure that there is an overlapping region within each adjacent azimuth block. | double | 1 |
| chirpReplicaSource | Chirp replica to use during processing. The extracted replica will be used if this parameter is set to "Extracted" and the IPF determines that the reconstructed replica is valid; otherwise, the nominal chirp will be used if this field is set to "Nominal" or the reconstructed replica is deemed invalid. | chirpSourceType | 1 |
| replicaThresholds | Thresholds used to assess the quality of the replica reconstruction and the PG product. | replicaThresholdParamsType | 1 |
| missingLinesThreshold | Threshold for setting the missingLinesSignificant flag in the output product annotations. This parameter ranges between 0 and 1 and specifies the percentage of missing lines to total lines [%]. For TOPS products, this threshold is relative to each individual swath, in other words the missingLinesSignificant flag is set to true if the number of missing lines in any of the swaths is above the missingLinesThreshold. | double | 1 |
| linesPerGapThreshold | This parameter specifies the number of consecutive missing lines in the input data which constitute a gap [lines]. | uint32 | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| missingGapsThreshold | Threshold for setting the significantGapsInInputData flag in the output product annotations. This parameter specifies the number of missing gaps permitted in the input data. For TOPS products, this threshold is relative to each individual swath, in other words the significantGapsInInputData flag is set to true if the number of missing gaps in any of the swaths is above the missingGapsThreshold. | uint32 | 1 |
| performInternalCalibrationFlag | Flag controlling the calculation of the internal calibration from the calibration pulses extracted from the downlink. If this flag is set to true then the internal calibration information will be calculated by the IPF using the calibration pulses extracted from the downlink. If this flag is set to false then the internal calibration information will not be calculated from the calibration pulses extracted from the downlink. In addition, if this flag is set to false, the values provided for chirpReplicaSource and pgSource will be ignored and set to "Nominal" and "Model" respectively. | bool | 1 |
| pgSource | PG source to use during processing. The PG derived from the extracted replica will be used if this parameter is set to “Extracted” and the IPF determines that the reconstructed replica is valid; otherwise, the pgModel from the AUX_INS data will be used if this field is set to “Model” or the reconstructed replica is deemed invalid. | pgSourceType | 1 |
| estimateNoiseEquivalentPowerFlag | Flag introduced starting from IPF V290 to manage the new L0N products format including also noise-equivalent echoes. If set to true, then the noise power is estimated also from the noise equivalent echoes; otherwise, if set to false only pure noise echoes are used. | bool | 1 |

**Table 4-10 Data Type - replicaThresholdParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| maxXCorrPulseIrw | Maximum allowable IRW (3-dB width) of the cross-correlation with the nominal replica [samples]. Used for setting the replicaReconstructionFailedFlag in the output product annotations. | double | 1 |
| maxXCorrPulsePslr | Maximum allowable PSLR of the cross correlation with the nominal replica [dB]. Used for setting the replicaReconstructionFailedFlag in the output product annotations. | double | 1 |
| maxXCorrPulseIslr | Maximum allowable ISLR of the cross correlation with the nominal replica [dB]. Used for setting the replicaReconstructionFailedFlag in the output product annotations. | double | 1 |
| maxPgAmpStdFraction | Maximum deviation from the mean allowed for the PG product amplitude, measured as a fraction of the standard deviation. Relative PG product validation shall fail if this value is exceeded. | float | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| maxPgPhaseStdFraction | Maximum deviation from the mean allowed for the PG product phase, measured as a fraction of the standard deviation. Relative PG product validation shall fail if this value is exceeded. | float | 1 |
| maxPgAmpError | Maximum deviation allowed for a PG product amplitude from the corresponding PG product model value. Absolute PG product validation shall fail if this value is exceeded. | float | 1 |
| maxPgPhaseError | Maximum deviation allowed for a PG product phase from the corresponding PG product model value [radians]. Absolute PG product validation shall fail if this value is exceeded. | float | 1 |
| maxNumInvalidPgValFraction | Maximum number of invalid PG product values allowed, expressed as a fraction of the total number of PG values. If the percentage of the invalid PG products does not exceed this value, then the invalid PG values will be discarded and only the valid PG values will be further used in the linear interpolation and application to the data. Otherwise if the percentage of the invalid PG products does exceed this value, then all the calculated PG product values will be discarded and replaced with the corresponding PG product model values. | float | 1 |

### 4.3.3 Doppler Centroid Processing Parameters

The Doppler centroid processing parameters section contains the parameters and options used to control and fine tune the Doppler centroid estimation within the IPF. Note that in the case of Doppler centroid estimation on IW or EW data, one estimate will be performed for every burst cycle. For WV data, one estimate will be performed for each vignette. For SM data, estimates will be performed periodically throughout the data depending on the internal configuration of the IPF. Figure 4-4 presents a high-level graphical overview of the Doppler centroid parameters and the subsequent tables describe the content of this DSR.

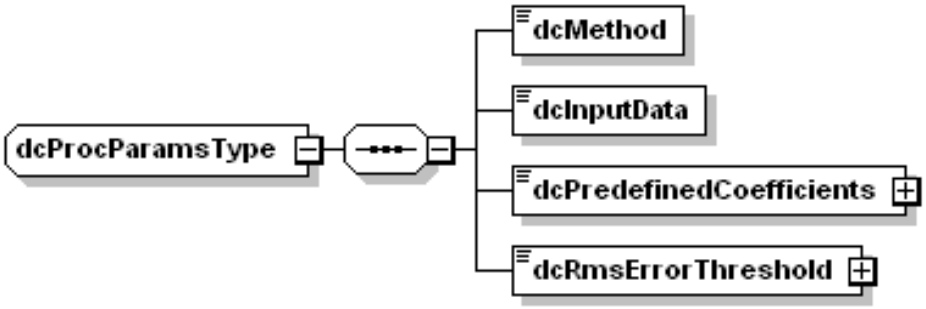


**Figure 4-4 Doppler Centroid Parameters Auxiliary Data**

**Table 4-11 Data Type - dcProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| dcMethod | Doppler centroid estimation method. Although the DCE from both orbit and attitude data and data analysis are always performed and the results provided in the output product annotations, this parameter is used to specify exactly which Doppler centroid estimation method to use during the image focusing. | dcMethodType | 1 |
| dcInputData | Type of input data used for Doppler centroid estimation. Options are “Raw” and “Range Compressed”. | dcInputDataType | 1 |
| dcPredefinedCoefficients | Pre-defined Doppler centroid coefficients. These Doppler centroid coefficients shall be used during processing if and only if the dcMethod element is set to "Pre-defined". This parameter is an array of floating point numbers separated by spaces. | floatCoefficientArray | 1 |
| dcRmsErrorThreshold | Doppler centroid estimation root mean squared (RMS) error threshold. If the RMS error of the Doppler centroid estimate from data is above this threshold they shall not be used during processing; instead, the Doppler centroid calculated from orbit and attitude shall be used, unless overridden by the dcMethod = Pre-defined. | float | 1 |

### 4.3.4 SLC Processing Parameters

The SLC processing parameters section contains the parameters and options used to control and fine tune the IPF during SLC processing. Figure 4-5 presents a high-level graphical overview of the SLC processing parameters and the subsequent tables describe the content of this DSR.

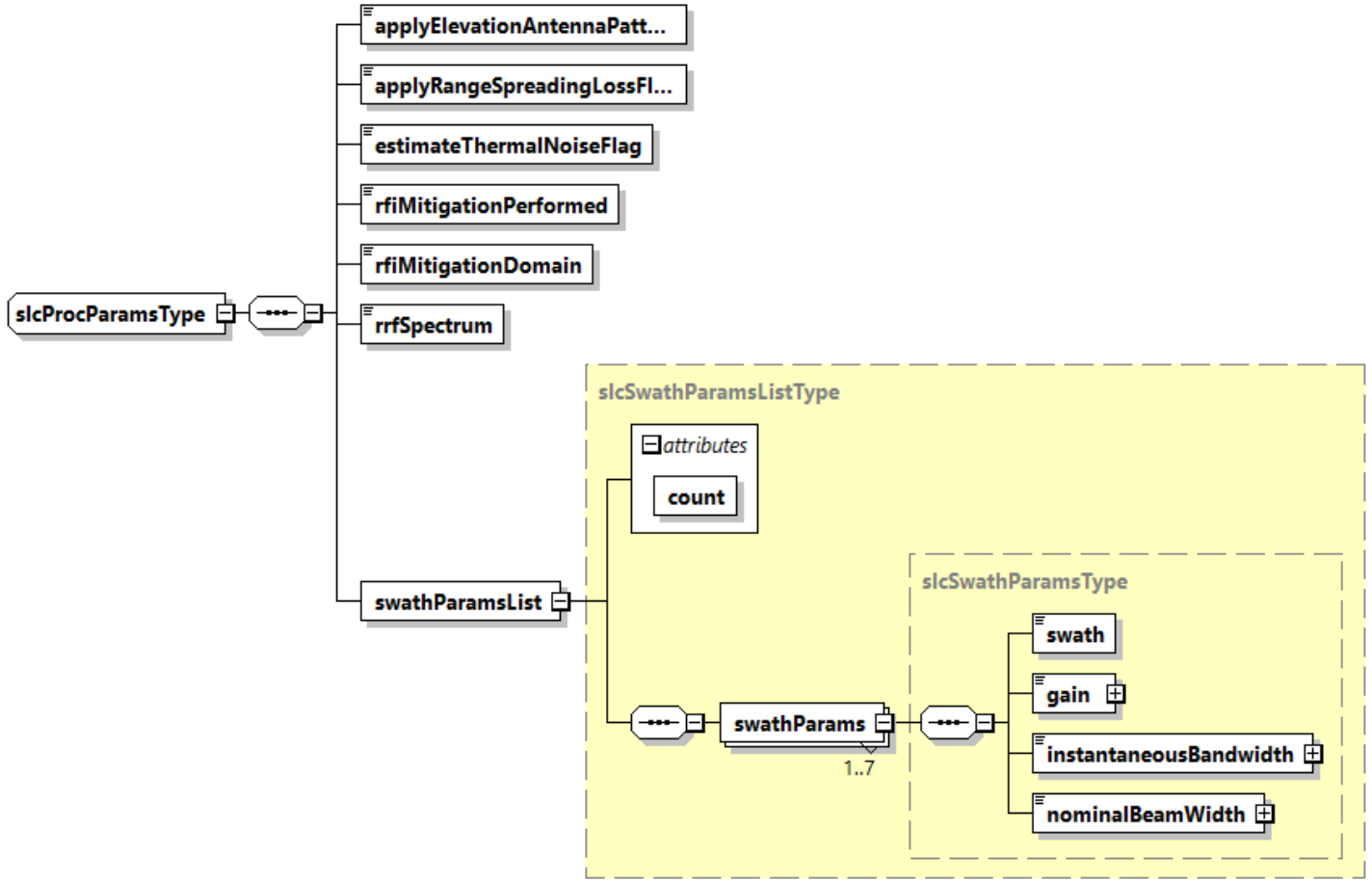


**Figure 4-5 SLC Parameters Auxiliary Data**

**Table 4-12 Data Type - slcProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| applyElevationAntennaPatternFlag | Elevation antenna pattern correction flag. True if the EAP is to be applied, false otherwise. | bool | 1 |
| applyRangeSpreadingLossFlag | Apply range spreading loss correction flag. True if the RSL is to be applied, false otherwise. | bool | 1 |
| estimateThermalNoiseFlag | Thermal noise estimation flag. True if thermal noise estimation is to be performed, false otherwise. | bool | 1 |
| rfiMitigationPerformed | Whether or not to perform the RFI mitigation step: if "Always" the RFI mitigation step is always performed; if "Never" the RFI mitigation step is not performed; if "BasedOnNoiseMeas" the preliminary RFI detection from noise measurements decides whether to perform RFI mitigation. | "Always", "Never", "BasedOnNoiseMeas" | 1 |
| rfiMitigationDomain | In case the RFI mitigation step is performed, this flag indicates in what domain it has to be applied: "Time", "Frequency", "TimeAndFrequency". | "Time", "Frequency", "TimeAndFrequency" | 1 |
| rrfSpectrum | The type of range matched filter to use during processing. "Unextended": range reference function is unextended in frequency domain; "Extended Flat": range reference function is extended and flat in frequency domain; and, "Extended Tapered": range reference function is extended and tapered in frequency domain. | rrfSpectrumType | 1 |
| swathParamsList | List of swath-dependent SLC processing parameters. There is one record per swath for each mode. | slcSwathParamsListType | 1 |

**Table 4-13 Data Type – slcSwathParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of parameter records in this list. There is one record per swath.* | *unsignedInt* | *required* |
| swathParams | Swath-dependent SLC processing parameters record indexed by swath. There will be one record per swath for each mode. For example, for S1 IW there will be 3 records, and for ASAR IM there will be 7 records. | slcSwathParamsType | 1 .. 7 |

**Table 4-14 Data Type - slcSwathParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Swath identifier. This parameter defines the swath to which this record applies. | swathType | 1 |
| gain | Product and polarisation specific gain. This parameter defines the gain that is applied to each output sample during azimuth processing. This parameter is an array of double precision floating point numbers separated by spaces, one for each of the possible polarisations in the following order:<br><gain count="4">HH HV VV VH</gain><br>Where the "HH", "HV", "VV", "VH" are replaced with the numerical gain applicable to that polarisation. | doubleCoefficientArray | 1 |
| instantaneousBandwidth | Azimuth instantaneous bandwidth [Hz]. This field is applicable only to IW and EW modes. | float | 1 |
| nominalBeamWidth | Nominal width of the beam [degrees]. This is the *nomBeamWidth* parameter from [A-11] and is used in the azimuth oversampling normalisation step. | double | 1 |

### 4.3.5 Post Processing Parameters

The post processing parameters section contains the parameters and options used to control and fine tune the IPF during image post processing including SLC, GRD and Quick-look processing. Figure 4-6 presents a high-level graphical overview of the post processing parameters and the subsequent tables describe the content of this DSR.

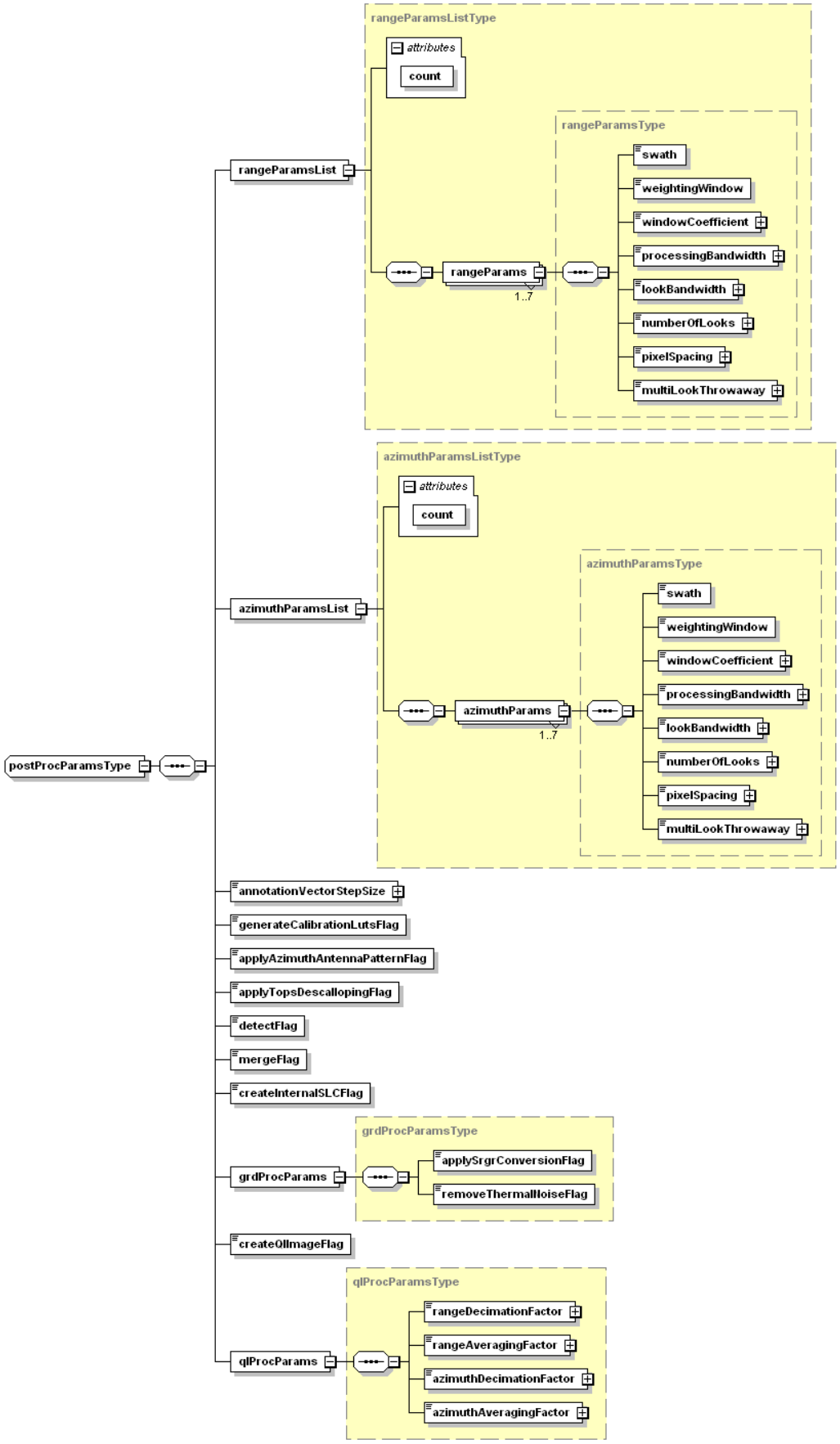


**Figure 4-6 Post Processing Parameters Auxiliary Data**

**Table 4-15 Data Type - postProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rangeParamsList | Range processing parameters. This list contains the swath-dependent auxiliary parameters required for range processing. The list contains a separate record for each swath, indexed using the applicable swath identifier. | rangeParamsListType | 1 |
| azimuthParamsList | Azimuth processing parameters. This list contains the swath-dependent auxiliary parameters required for azimuth processing. The list contains a separate record for each swath, indexed using the applicable swath identifier. | azimuthParamsListType | 1 |
| annotationVectorStepSize | The decimation factor used on the calibration and noise annotation vectors when written to the output product. Inside the IPF, each annotation vector could have a point for every range sample. To reduce product size, only points every annotationVectorStepSize are written to the output calibration and noise annotation vectors. | uint32 | 1 |
| generateCalibrationLutsFlag | Flag to control the generation of the absolute calibration LUTs. True if the calibration LUTs are to be created, false otherwise. | bool | 1 |
| applyAzimuthAntennaPatternFlag | Azimuth Antenna Pattern Flag. True if the AAP is to be applied, false otherwise. | bool | 1 |
| applyTopsDescallopingFlag | Perform de-scalloping for TOPS by applying the Azimuth Antenna Element Pattern. True if TOPS de-scalloping is to be performed by applying the AAEP, false otherwise. This flag is applicable only to IW and EW modes and is ignored for all others. | bool | 1 |
| detectFlag | True to detect power and square root extract measurement data, false otherwise. This flag should only be set to true for GRD products. | bool | 1 |
| mergeFlag | True to merge the swaths of output data and annotations, false otherwise. Only valid for IW and EW, not applicable to other modes. This flag should only be set to true for GRD products. | bool | 1 |
| createInternalSLCFlag | True if the output of post-processing is an internal SLC product, in which case post-processing will pass the input SLC product through without applying further processing. In the case where the final output product is an external L1 product, this field shall be set to false. In the case where the final output product is an external L2 product, this field shall be set to true. | bool | 1 |
| grdProcParams | GRD processing auxiliary parameters. This record contains the auxiliary parameters required during GRD image processing. | grdProcParamsType | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| createQlImageFlag | Create Quick-look image flag. This flag controls the creation of a Quick-look image. It is set to true if a Quick-look image should be created; or, false otherwise. | bool | 1 |
| qlProcParams | Quick-look processing auxiliary parameters. This record contains the auxiliary parameters required during Quick-look image processing. This structure need only be present if the createQlImageFlag is true. | qlProcParamsType | 1 |

**Table 4-16 Data Type – rangeParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of parameter records in this list. There is one record per swath for each instrument mode.* | *unsignedInt* | *required* |
| rangeParams | This record contains the set of auxiliary parameters required during range processing. For each product type, there is one record for each applicable swath within the product. There will be one record per swath for each mode. For example, for S1 IW there will be 3 records, and for ASAR IM there will be 7 records. | rangeParamsType | 1 .. 7 |

**Table 4-17 Data Type - rangeParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Swath identifier. This parameter defines the swath to which this record applies. | swathType | 1 |
| weightingWindow | Name of the weighting window to use during processing. | weightingWindowType | 1 |
| windowCoefficient | Value of the weighting window coefficient to use during processing. | double | 1 |
| processingBandwidth | Total processing bandwidth [Hz]. This parameter defines the bandwidth (Bw) to use during processing. For range, the processing bandwidth (Bw) must be 0 < Bw <= pulse Bw, for azimuth the processing bandwidth (Bw) must be 0 < Bw <= PRF. | double | 1 |
| lookBandwidth | Look bandwidth [Hz]. This parameter defines the bandwidth to use for each look during processing. | double | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| numberOfLooks | Number of looks. This parameter defines the number of looks to use during multi-look processing. | uint32 | 1 |
| pixelSpacing | Spacing between pixels in the output image [m]. | double | 1 |
| multiLookThrowaway | Number of ground range samples to be discarded at both edges. | int32 | |

**Table 4-18 Data Type - azimuthParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of parameter records in this list. There is one record per swath for each instrument mode.* | *unsignedInt* | *required* |
| azimuthParams | This record contains the set of auxiliary parameters required during azimuth processing. For each product type, there is one record for each applicable swath within the product. There will be one record per swath for each mode. For example, for S1 IW there will be 3 records, and for ASAR IM there will be 7 records. | azimuthParamsType | 1 .. 7 |

**Table 4-19 Data Type - azimuthParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Swath identifier. This parameter defines the swath to which this record applies. | swathType | 1 |
| weightingWindow | Name of the weighting window to use during processing. | weightingWindowType | 1 |
| windowCoefficient | Value of the weighting window coefficient to use during processing. | double | 1 |
| processingBandwidth | Total processing bandwidth [Hz]. This parameter defines the bandwidth (Bw) to use during processing. For range, the processing bandwidth (Bw) must be 0 < Bw <= pulse Bw, for azimuth the processing bandwidth (Bw) must be 0 < Bw <= PRF. | double | 1 |
| lookBandwidth | Look bandwidth [Hz]. This parameter defines the bandwidth to use for each look during processing. | double | 1 |
| numberOfLooks | Number of looks. This parameter defines the number of looks to use during multi-look processing. | uint32 | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| pixelSpacing | Spacing between pixels in the output image [m].<br>Note: this field do not drive the data final spacing, but this is just a post-processing parameter only valid for GRD products. For SLC products the output pixel spacing is driven by an IPF internal configuration parameter. | double | 1 |
| multiLookThrowaway | Number of ground range samples to be discarded at both edges. | int32 | 1 |

**Table 4-20 Data Type - grdProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| applySrgrConversionFlag | Slant Range to Ground Range (SRGR) conversion flag. True if SRGR conversion is to be performed, false otherwise. | bool | 1 |
| removeThermalNoiseFlag | Thermal noise removal flag. True if thermal noise removal is to be performed, false otherwise. | bool | 1 |

**Table 4-21 Data Type – qlProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rangeDecimationFactor | Range decimation factor for the image. | uint32 | 1 |
| rangeAveragingFactor | Range averaging factor for the image. | uint32 | 1 |
| azimuthDecimationFactor | Azimuth decimation factor for the image. | uint32 | 1 |
| azimuthAveragingFactor | Azimuth averaging factor for the image. | uint32 | 1 |

### 4.3.6 Application LUT Processing Parameters

The application LUT processing parameters section contains the definition of the application scaling LUTs that are optionally used during processing to compensate for near-to-far range radiometric variances and optimise the use of the available dynamic range in the output product. Figure 4-7 presents a high-level graphical overview of the application LUT processing parameters and the subsequent tables describe the content of this DSR.

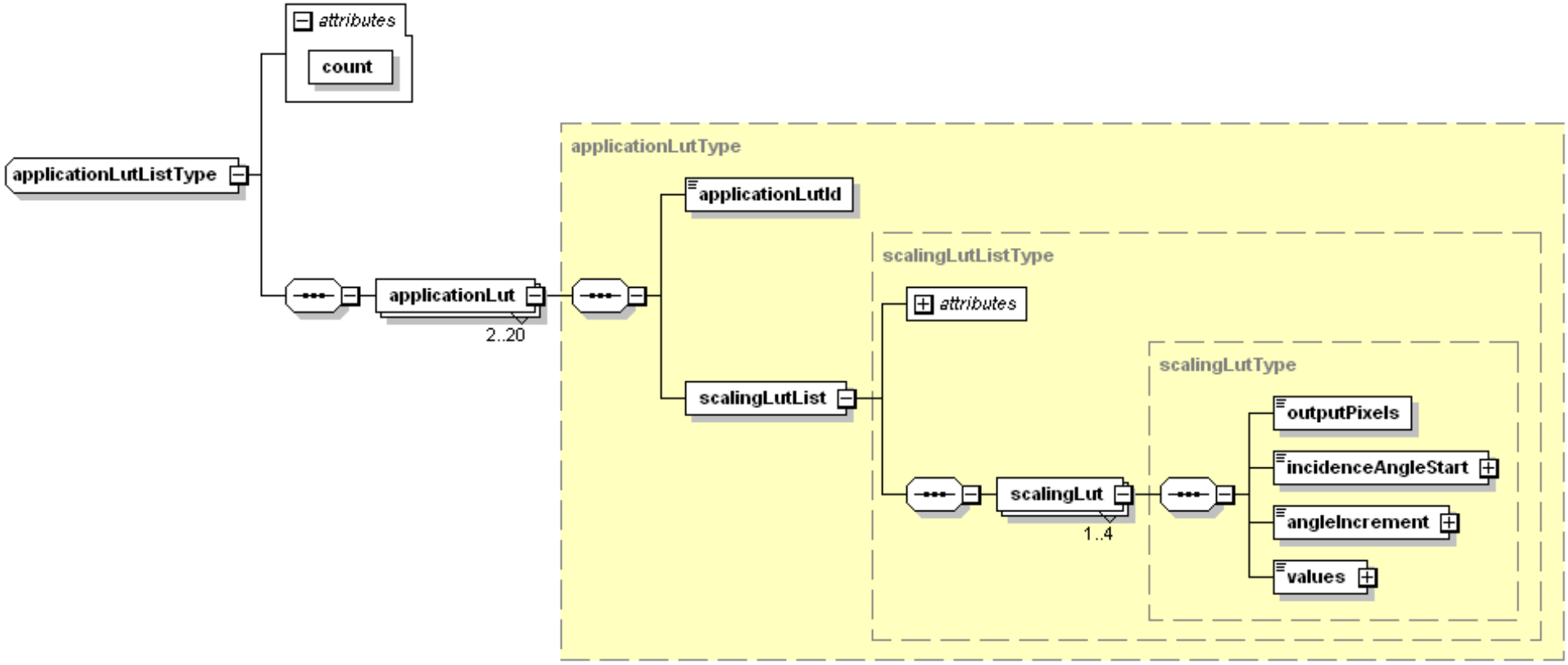


**Figure 4-7 Application LUT Processing Parameters Auxiliary Data**

**Table 4-22 Data Type - applicationLutListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *The number of applicationLut records within the list.* | *unsignedInt* | *Required* |
| applicationLut | Application LUT record. This element contains the information required to identify the application scaling LUT specified in the Job Order. The ICD [A-5] currently defines four default application scaling LUTs, one for each mode. However, more can be added, so long as the applicationLutId matches the Application_LUT processing parameter in the Job Order. | applicationLutType | 2 .. 20 |

**Table 4-23 Data Type - applicationLutType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| applicationLutId | Name of this application scaling LUT. | string | 1 |
| scalingLutList | List of application scaling LUTs for this applicationLutId. There is one entry for each output pixel type. | scalingLutListType | 1 |

**Table 4-24 Data Type - scalingLutListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *The number of scalingLut records within the list.* | *unsignedInt* | *Required* |
| scalingLut | Application scaling LUT record. This record provides the scaling LUT for products in the format specified by the outputPixels field. | scalingLutType | 1 .. 4 |

**Table 4-25 Data Type - scalingLutType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| outputPixels | Pixel format. Application scaling LUTs are specific to output pixel data type. This field specifies the output pixel type to which this LUT applies. | outputPixelsType | 1 |
| incidenceAngleStart | Incidence angle of the first value in the LUT [degrees]. | double | 1 |
| angleIncrement | Step size of the incidence angle between each value [degrees]. | double | 1 |
| values | Application LUT values. This element is a vector containing attribute “count” floating point values separated by spaces [linear]. | floatArray | 1 |

# 5 CALIBRATION AUXILIARY DATA

## 5.1 Overview

The calibration auxiliary product includes information required for relative and absolute product calibration. The information in this file is calculated during the mission calibration activities and used by the IPF to calibrate the L1 image products.

New versions of this auxiliary product will be provided when the calibration information needs to be modified due to changes in the instrument characteristics.

### 5.1.1 Product Structure

Table 5-1 describes the physical structure and data file components included in the calibration auxiliary product as well as a nominal estimated size for each.

**Table 5-1 Sentinel-1 Calibration Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| MMM_AUX_CAL_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| mmm-aux-cal.xml | Auxiliary Data | 1 | 420 KB |
| support/ | Directory | 1 | N/A |
| s1-aux-cal.xsd | Representation Data | 1 | 10 KB |
| s1-object-types.xsd | Representation Data | 1 | 48 KB |

### 5.1.2 Naming Standard

Naming for the calibration auxiliary product follows the standard defined in Section 3.2.1 and naming of the component files within the calibration auxiliary product follows the standard defined in Section 3.2.2.

## 5.2 Data File Definition

The schema file for the definition of the calibration auxiliary file is specified in Appendix A2.

## 5.3 Data File Content

Figure 5-1 presents a graphical view of the calibration auxiliary data contained in the auxiliary data file mmm-aux-cal.xml, and a detailed tabular view is presented in the tables below.

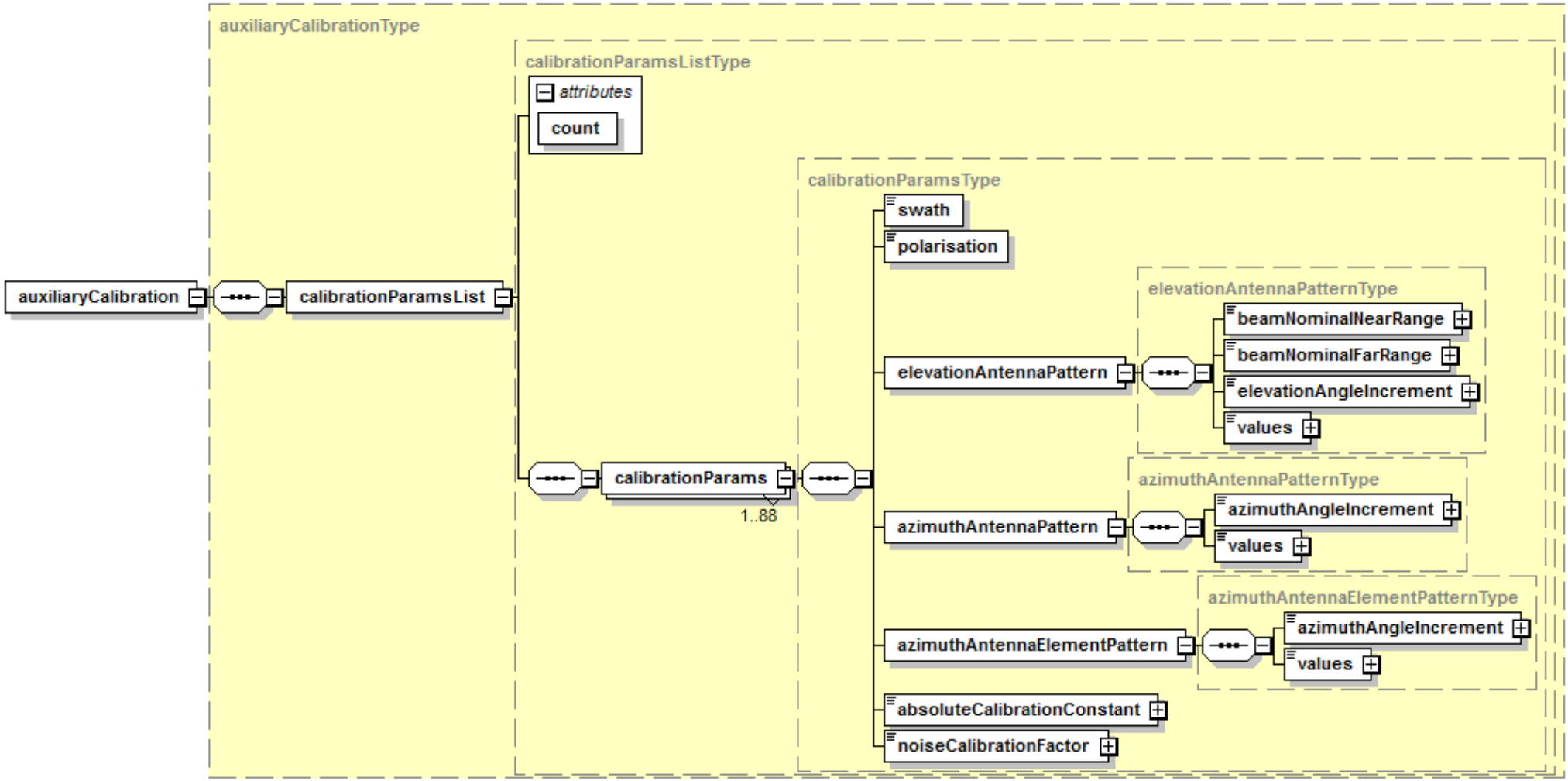


**Figure 5-1 Calibration Auxiliary Data**

**Table 5-2 Element - auxiliaryCalibration**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| calibrationParamsList | List of calibration parameter records. | calibrationParamsListType | 1 |

**Table 5-3 Data Type - calibrationParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of calibration parameter records in the list.* | *unsignedInt* | *required* |
| calibrationParams | Calibration parameter record. There may be up to one record per swath (23 nominal swaths) per polarisation (4 polarisation combinations for SM, IW, EW, EN and AN, 2 for WV) for a maximum total of 88 records. | calibrationParamsType | 1 .. 88 |

**Table 5-4 Data Type - calibrationParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Canonical name of the swath to which this set of calibration parameters applies. The swath and polarisation are used to index the applicable calibrationParams record. | swathType | 1 |
| polarisation | Polarisation to which this set of calibration parameters applies. The polarisation and swath are used to index the applicable calibrationParams record. | polarisationType | 1 |
| elevationAntennaPattern | Two way elevation antenna pattern parameters. The EAPs are used to correct the corresponding radiometric variation of the data in the range direction. The EAPs are also used for the estimation and removal of the thermal noise level. | elevationAntennaPatternType | 1 |
| azimuthAntennaPattern | Two way azimuth antenna pattern (AAP) parameters. | azimuthAntennaPatternType | 1 |
| azimuthAntennaElementPattern | Two way azimuth antenna element pattern values. The AAEP maps azimuth angles to gain power and is used during de-scalloping of TOPSAR data. The AAEP is specific to IW and EW modes and is ignored for all others. | azimuthAntennaElementPattern Type | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| absoluteCalibrationConstant | Absolute calibration constant value to apply during processing. Although the structure of the file allows for a unique value per swath and polarisation the value of this field must be the same for all swaths and polarisations within the mode. The calibration constant C0 (described in the Sentinel-1 SAR Instrument Calibration and Characterisation Plan) should be merged into this field to achieve an overall gain value. | double | 1 |
| noiseCalibrationFactor | Noise calibration factor used in the estimation of the thermal noise. | double | 1 |

**Table 5-5 Data Type - elevationAntennaPatternType**

| Element | Description | Data Type | Cardinality |
|---|---|---|---|
| beamNominalNearRange | Elevation angle of the nominal near range extent of the beam [degrees]. Note that this parameter is not currently used by the IPF. | double | 1 |
| beamNominalFarRange | Elevation angle of the nominal far range extent of the beam [degrees]. Note that this parameter is not currently used by the IPF. | double | 1 |
| elevationAngleIncrement | Elevation angle increment [degrees]. This parameter defines the step size between the pattern values in the two way elevation antenna pattern. | double | 1 |
| values | Two-way complex antenna elevation pattern values. The centre value in the vector corresponds to the referenceAntennaAngle in the Roll Steering parameters described in the Instrument auxiliary file (the vector must contain an odd number of complex values), and the values before and after the centre value correspond to steps of elevationAngleIncrement from the centre value. The pattern contains attribute "count" complex floating point values separated by spaces in the order I Q I Q I Q... The first value in the antenna pattern vector corresponds to –((count - 1)/2) * elevationAngleIncrement degrees, and the last value corresponds to +((count - 1)/2) * elevationAngleIncrement degrees. The complex values in this vector are applied to the complex image data as: $ComplexDataCorrected_{(x,y)} = ComplexData_{(x,y)} / (\text{sqrt}(\text{abs}(ComplexEAP_{(x,y)}))*\text{phase}(ComplexEAP_{(x,y)}))$ | complexArray | 1 |

**Table 5-6 Data Type - azimuthAntennaPatternType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| azimuthAngleIncrement | Azimuth angle increment [degrees]. This parameter defines the step size between the values in the two way azimuth antenna pattern. | double | 1 |
| values | Two way azimuth antenna pattern values [dB]. The centre value of the vector corresponds to 0 degrees (the vector must contain an odd number of values), and the values before and after the centre value correspond to steps of azimuthAngleIncrement degrees from the centre value. The pattern contains attribute "count" floating point values separated by spaces. The first value in the antenna pattern vector corresponds to –((count - 1)/2) * azimuthAngleIncrement degrees, and the last value corresponds to +((count - 1)/2) * azimuthAngleIncrement degrees. | floatArray | 1 |

**Table 5-7 Data Type - azimuthAntennaElementPatternType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| azimuthAngleIncrement | Azimuth angle increment [degrees]. This parameter defines the step size between the values in the two way azimuth antenna element pattern. | double | 1 |
| values | Two way azimuth antenna element pattern values [dB]. The centre value of the vector corresponds to 0 degrees (the vector must contain an odd number of values), and the values before and after the centre value correspond to steps of azimuthAngleIncrement degrees from the centre value. The pattern contains attribute "count" floating point values separated by spaces. The first value in the antenna pattern vector corresponds to –((count - 1)/2) * azimuthAngleIncrement degrees, and the last value corresponds to +((count - 1)/2) * azimuthAngleIncrement degrees. | floatArray | 1 |

# 6 INSTRUMENT AUXILIARY DATA

## 6.1 Overview

The instrument auxiliary product includes information about the settings and state of the SAR instrument that is required by the IPF for processing raw data products into L1 image products. Information required for data decompression and ISP decoding is also contained in the instrument auxiliary product.

New versions of this auxiliary product will be provided when the information needs to be modified due to changes in the instrument settings. For example, changes to the imaging sequence for a beam mode or the addition of a new beam mode.

### 6.1.1 Product Structure

Table 6-1 describes the physical structure and data file components included in the instrument auxiliary product as well as a nominal estimated size for each.

**Table 6-1 Sentinel-1 Instrument Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| MMM_AUX_INS_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| mmm-aux-ins.xml | Auxiliary Data | 1 | 115 KB |
| support/ | Directory | 1 | N/A |
| s1-aux-ins.xsd | Representation Data | 1 | 25 KB |
| s1-object-types.xsd | Representation Data | 1 | 48 KB |

### 6.1.2 Naming Standard

Naming for the instrument auxiliary product follows the standard defined in Section 3.2.1 and naming of the component files within the instrument auxiliary product follows the standard defined in Section 3.2.2.

## 6.2 Data File Definition

The schema file for the definition of the instrument auxiliary file is specified in Appendix A3.

## 6.3 Data File Content

Figure 6-1 presents a graphical view of the instrument auxiliary data contained in the auxiliary data file mmm-aux-ins.xml, and a detailed tabular view is presented in the following tables.

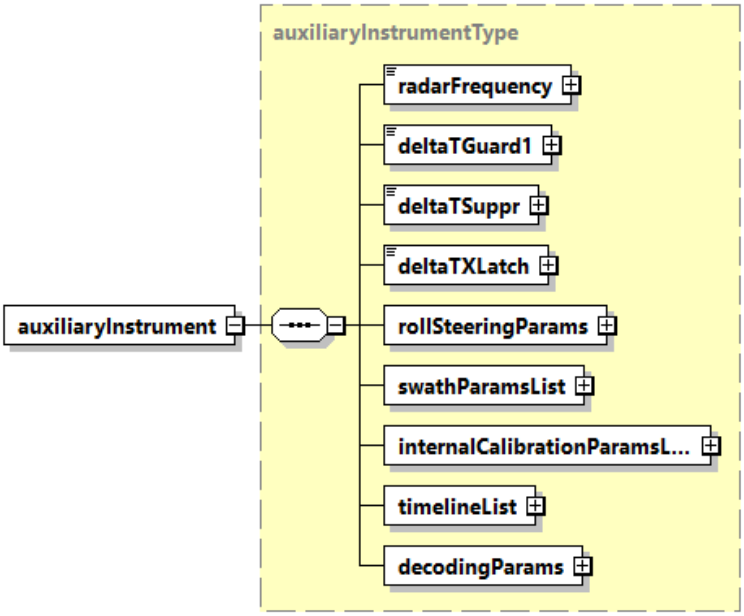


**Figure 6-1 Instrument Auxiliary Data**

**Table 6-2 Element - auxiliaryInstrument**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| radarFrequency | Radar frequency [Hz]. | double | 1 |
| deltaTGuard1 | DeltaT Guard 1 parameter from the instrument radar database [s]. This parameter is used to calculate the internal time delay of the extracted reconstructed replicas. | double | 1 |
| deltaTSuppr | DeltaT Suppr parameter from the SSPPDU document [s]. This parameter is used to calculate the times within the PRI for the echo, calibration and noise data. | double | 1 |
| deltaTXLatch | The time difference between the TX pulse and the latch time as annotated in the radar database [s]. This parameter is used to apply the instrument timing correction. | double | 1 |
| rollSteeringParams | Parameters related to the roll steering law for the instrument. The roll steering law defines the off nadir pointing of the antenna mechanical bore sight versus time. | rollSteeringParamsType | 1 |
| swathParamsList | Swath parameters list. This element contains a list of swath dependent instrument parameters. | swathParamsListType | 1 |
| internalCalibrationParamsList | Internal calibration parameters list. This element contains a list of swath/polarisation channel dependent instrument parameters. | internalCalibrationParamsListType | 1 |
| timelineList | Timeline list. This element contains a list of records that describe the expected packet transmission sequence for each of the Sentinel-1 SAR modes. | timelineListType | 1 |
| decodingParams | Raw data decoding parameters. This DSR contains the raw data decoding tables and parameters that the IPF requires to perform image processing. | decodingParamsType | 1 |

### 6.3.1 Roll Steering Auxiliary Parameters

The roll steering parameters define the roll steering law for the instrument. The roll steering law defines the off nadir pointing of the antenna mechanical bore sight versus time, as described in Document R-4.

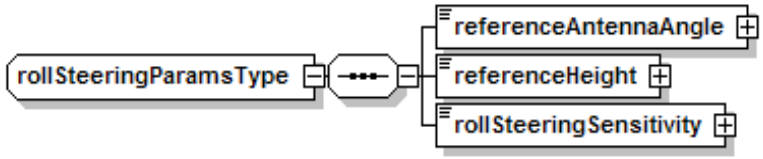


**Figure 6-2 Roll Steering Auxiliary Parameters**

**Table 6-3 Data Type - rollSteeringParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| referenceAntennaAngle | Antenna bore sight off nadir angle at the referenceHeight [degrees]. | double | 1 |
| referenceHeight | Satellite height at which the instrument elevation angle is aligned with the referenceAntennaAngle [m]. | double | 1 |
| rollSteeringSensitivity | Sensitivity of the roll steering versus height [degrees/m]. | double | 1 |

### 6.3.2 Swath Instrument Auxiliary Parameters

The swath instrument auxiliary parameters contain parameters that describe key characteristics of each swath within the SAR imaging modes.

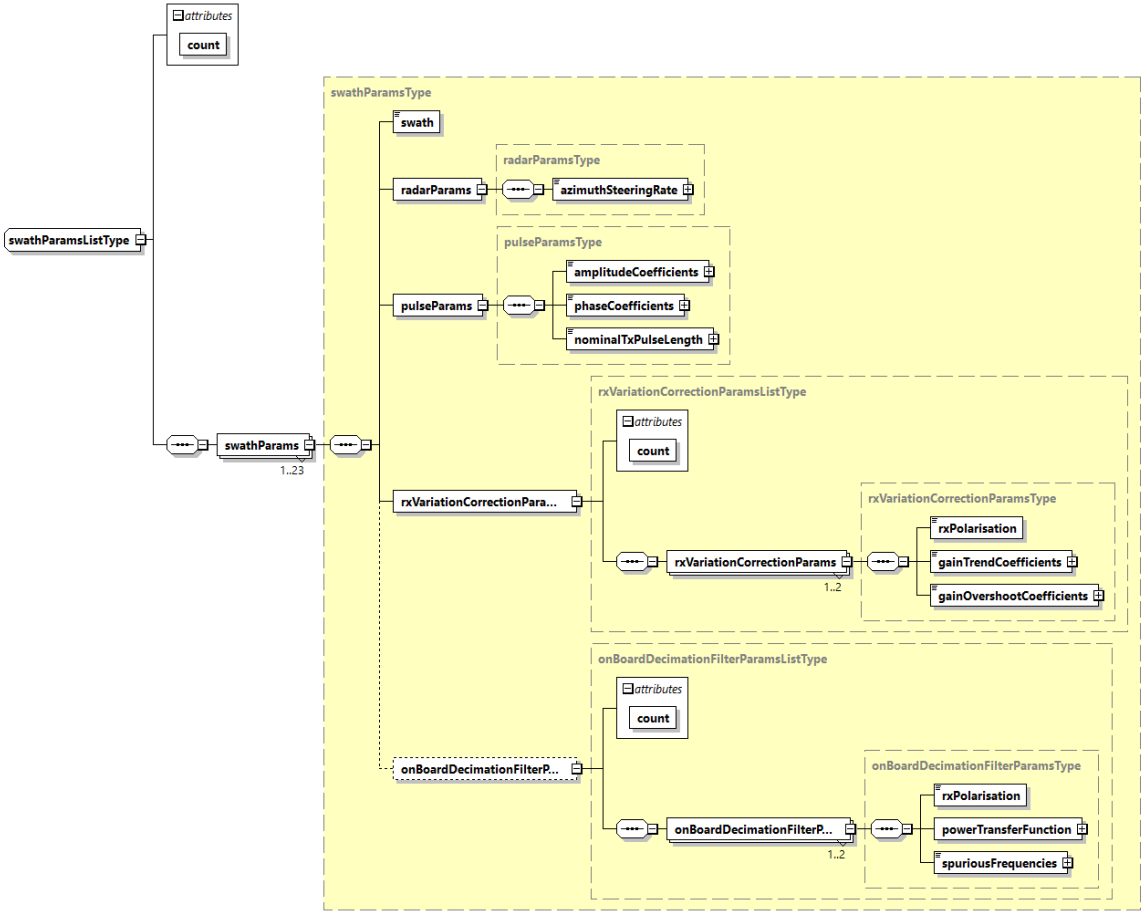


**Figure 6-3 Swath Instrument Auxiliary Parameters**

**Table 6-4 Data Type - swathParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of swathParams records in the list.* | *unsignedInt* | *required* |
| swathParams | Swath Parameters. This record contains swath-dependent parameters related to the instrument. There may be up to one record per swath for a maximum total of 23 records. | swathParamsType | 23 .. 128 |

**Table 6-5 Data Type - swathParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Canonical name of the swath to which this set of swathParams applies. | swathType | 1 |
| radarParams | Radar parameters. This DSR contains information related to the SAR instrument. | radarParamsType | 1 |
| pulseParams | Replica pulse parameters. This DSR contains the characteristics for the nominal imaging replica pulse within this swath. | pulseParamsType | 1 |
| rxVariationCorrectionParamsList | List of the receive variation correction parameters used to correct the gain variation across the receive window. | rxVariationCorrectionParamsListType | 1 |
| onBoardDecimationFilterParamsList | List of the on-board decimation filter parameters. | onBoardDecimationFilterParamsListType | 0-1 |

**Table 6-6 Data Type - radarParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| azimuthSteeringRate | TOPSAR azimuth steering rate [degrees/s]. This field is only relevant for IW and EW swaths and is set to 0.0 for SM and WV swaths. | double | 1 |

**Table 6-7 Data Type - pulseParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| amplitudeCoefficients | Pulse amplitude coefficients of the nominal imaging chirp replica. The amplitude coefficients are always set to the values:<br>C0 = 1.0<br>C1 = 0.0<br>C2 = 0.0<br>C3 = 0.0 | floatCoefficientArray | 1 |
| phaseCoefficients | Pulse phase coefficients of the nominal imaging chirp replica. The values of the phase coefficients are calculated as:<br>C0 = 0.0<br>C1 = TXPSF – TXPRR * (-TXPL/2)<br>C2 = TXPRR/2<br>C4 = 0.0<br>Where:<br>• TXPSF = Transmit pulse start frequency [Hz].<br>• TXPRR = Transmit pulse ramp rate [Hz/s].<br>• TXPL = Transmit pulse length [s]. | floatCoefficientArray | 1 |
| nominalTxPulseLength | Nominal transmit pulse length [s]. This parameter is used by the pre-processor and the DCE and SLC processors if it is greater than zero and smaller than the Tx Pulse Length (TXPL) extracted from the ISP headers. The nominal transmit pulse length can be set such that the chirp processed bandwidth is small enough to filter out the spurious signals at ±37.5 MHz.<br>Note: The IPF will ignore this parameter and use the extracted TXPL when the parameter is set to a value less than or equal to 0. | double | 1 |

**Table 6-8 Data Type – rxVariationCorrectionParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of rxVariationCorrectionParams records in the list.* | *unsignedInt* | *required* |
| rxVariationCorrectionParams | Receive variation correction parameters record. This record contains the coefficients used to correct the gain variation across the receive window. There may be up to one record per receive polarisation for a maximum total of 2 records. | rxVariationCorrectionParamsType | 1 .. 2 |

**Table 6-9 Data Type – rxVariationCorrectionParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rxPolarisation | Polarisation to which this set of receive correction parameters applies. "H" or "V". | rxPolarisationType | 1 |
| gainTrendCoefficients | Gain trend correction coefficients. Coefficient order is [p0 p1 p2 p3 p4 q0 q1 q2]. Usage: $GV1dB(t1) = (p_4*t_1^4 + p_3* t_1^3 + p_2* t_1^2 + p_1* t_1 + p_0) / (q_2*t_1^2 + q_1*t_1 + q_0)$ | doubleCoeffientArray | 1 |
| gainOvershootCoefficients | Gain overshoot correction coefficients. Coefficient order is [a b c d phi]. Usage: $GV2dB(ts) = a * \exp(-b*t_s) * \sin(d*t_s+phi) + c$ | doubleCoeffientArray | 1 |

**Table 6-10 Data Type - onBoardDecimationFilterParamsList**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| onBoardDecimationFilterParams | On-board decimation filter record. This record contains the power-transfer function of the filter and its known spurious frequencies. | onBoardDecimationFilterParamsType | 1-2 |
| count | Number of onBoardDecimationFilterParams records in the list. | unsignedInt | 1 |

**Table 6-11 Data Type - onBoardDecimationFilterParams**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rxPolarisation | Polarisation to which this set of parameters applies. "H" or "V". | “H”, “V” | 1 |
| powerTransferFunction | Power transfer function characterization of the on-board decimation filter. | onBoardDecimationFilterPowTransferFun | 1 |
| spuriousFrequencies | List of the known spurious frequencies. | floatArray | 1 |

**Table 6-12 Data Type - powerTransferFunction**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| frequencyIncrement | The frequency increment. | float | 1 |
| values | Power transfer function coefficients. | floatArray | 1 |

### 6.3.3 Internal Calibration Instrument Auxiliary Parameters

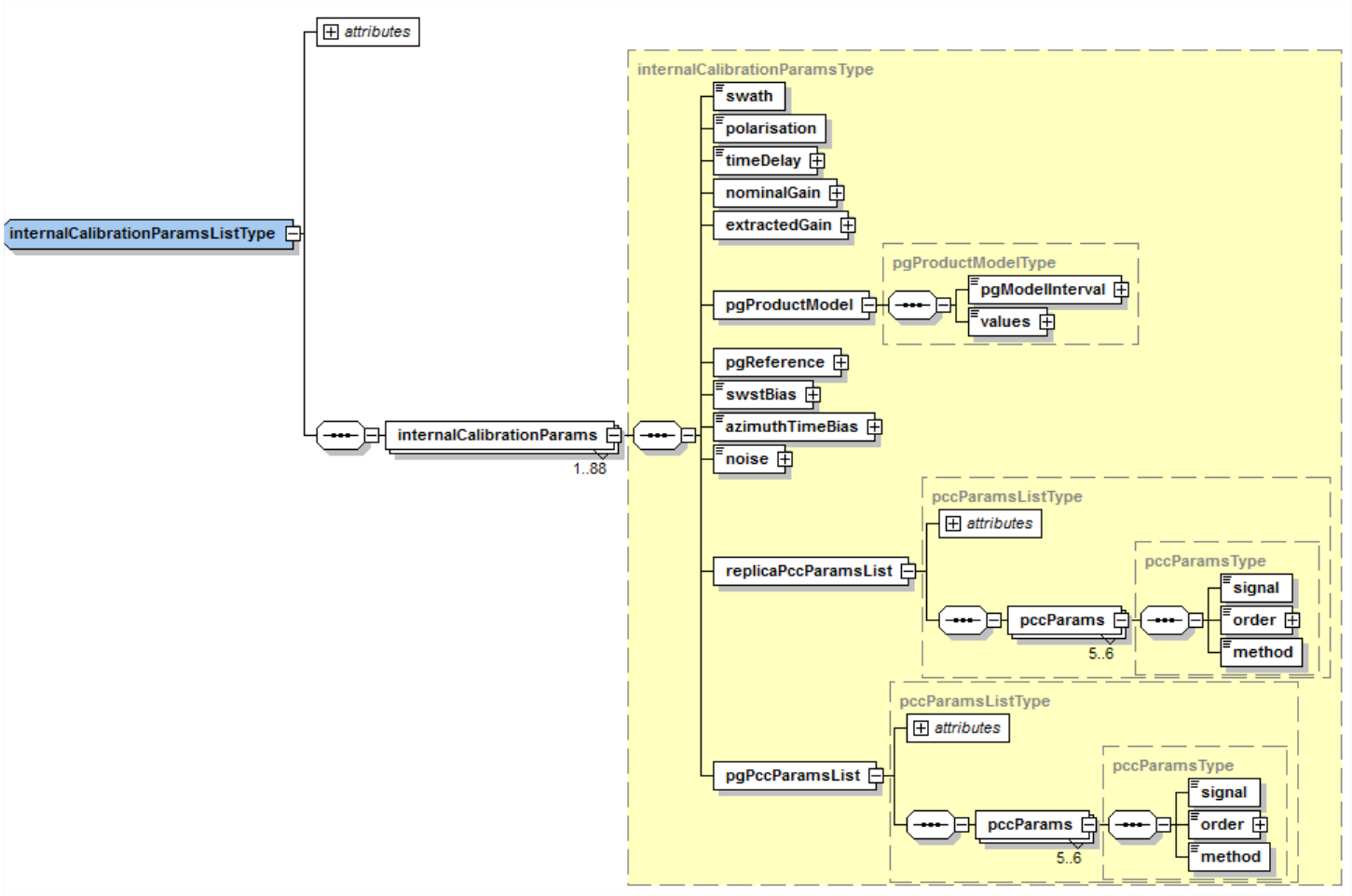


**Figure 6-4 Internal Calibration Instrument Auxiliary Parameters**

**Table 6-13 Data Type - internalCalibrationParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of internalCalibrationParams records in the list.* | *unsignedInt* | *required* |
| internalCalibrationParams | Internal calibration instrument parameters. This record contains swath/polarisation channel dependent parameters related to the instrument. There may be up to one record per swath (23 nominal swaths) per polarisation (4 polarisation combinations for SM, IW, EW, EN and AN, 2 for WV) for a maximum total of 88. | internalCalibrationParamsType | 1 .. 88 |

**Table 6-14 Data Type - internalCalibrationParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Canonical name of the swath to which this set of internal calibration parameters applies. The swath and polarisation are used to index the applicable internalCalibrationParams record. | swath | 1 |
| polarisation | Polarisation to which this set of internal calibration parameters applies. The polarisation and swath are used to index the applicable internalCalibrationParams record. | polarisationType | 1 |
| timeDelay | Internal time delay [s] to be applied to the reference chirp used for range processing. The IPF only uses this parameter when the time delay cannot be derived from the extracted PG replicas due to invalid (poor quality) reconstructed PG replicas. Nominally, the time delay is estimated as the average internal time delay from the all of the extracted PG replicas and it compensates for the dual polarisation mis-registration. | double | 1 |
| nominalGain | Complex gain to be applied to the range match filter when the nominal chirp is used. It is used to compensate for the amplitude and phase differences between the two channels in dual polarisation data. | complex | 1 |
| extractedGain | Complex gain to be applied to the range match filter when the extracted replica is used. It is used to compensate for the amplitude difference between the two channels in dual polarisation data. Phase difference compensation is not done when the extracted replica is used since the phase difference is already embedded in the extracted replicas therefore the phase value should always be set to 0 for the extracted replica case. | complex | 1 |
| pgProductModel | Modeled PG products. The model is relative to the ascending node of the current orbit. | pgProductModelType | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| pgReference | Reference absolute PG value that is defined by offline analysis of the acquired data. This PG reference value $PG_{ref}$ is divided by each PG value estimated from the calibration pulses – $PG_{cal}(t)$ – to obtain the relative PG value $PG(t)$ that can be compared to the PG model value $PG_{model}(t)$ in order to perform the relative and absolute PG validation described in [A-11]. This $PG_{ref}$ value is the same dimension (or order) as the PG value $PG_{cal}(t)$ estimated from the PG calibration pulses.<br>Therefore:<br>$PG(t) = PG_{ref} / PG_{cal}(t) \approx PG_{model}(t)$<br>See also pgProductModelType->values | complex | 1 |
| swstBias | SWST bias [s]. | double | 1 |
| azimuthTimeBias | Azimuth time bias [s].<br>Flag introduced starting from IPF V290 in order to have the possibility to correct for systematic time biases along azimuth direction (similarly to what done along range through SWST bias). | double | 1 |
| noise | Nominal noise value if noise used in processing if no noise value can be calculated from the downlink. | double | 1 |
| replicaPccParamsList | List of PCC decoding control parameters for the extracted replicas at nominal imaging bandwidth. | pccParamsListType | 1 |
| pgPccParamsList | List of PCC decoding control parameters for the PG replicas at 100 MHz bandwidth. | pccParamsListType | 1 |

**Table 6-15 Data Type - pgProductModelType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| pgModelInterval | Interval between adjacent PG model values in the list [s]. | double | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| values | Array of modeled complex PG model values. The pattern contains attribute "count" complex floating point values separated by spaces. The first value in the array corresponds to the time at the ascending node of the current orbit. The PG model values for times that fall between the points in the model are obtained by linear interpolation between the two nearest points.<br>Each PG value $PG_{model}(t)$ in this model is a relative value that captures the expected PG drift as a function of time. Each value in the model is the same dimension (or order) as $PG_{ref}$ / $PG_{cal}(t)$; therefore, the values in the model are expected, ideally, to be close to (1.0, 0.0).<br>See also internalCalibrationParamsType->pgReference | complexArray | 1 |

**Table 6-16 Data Type - pccParamsListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of pccParams records in the list.* | *unsignedInt* | *required* |
| pccParams | PCC decoding parameters for controlling the order and way in which calibration pulses are decoded during processing. There are a minimum of 5 entries in the list, one for each nominal calibration pulse, and a maximum of 6 entries in the list for the transmit H polarisation which includes an additional isolation pulse. | pccParamsType | 5 .. 6 |

**Table 6-17 Data Type - pccParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| signal | Signal type. | signalType | 1 |
| order | PCC pulse selection order. This is a list of integers separated by spaces that defines the order in which the pulses are combined using the method below. Examples:<br><!-- Select the 20th pulse first and the 19th pulse second --><br><order count="2">20 19</order><br><!-- Select pulses 13 through 20 in ascending order --><br><order count="8">13 14 15 16 17 18 19 20</order> | intArray | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| method | Method to use to combine the calibration pulses selected by the order above.<br>The PCC2 method subtracts the pulses in order and averages over the number of pulses.<br>The Average method adds the pulses in order and averages over the number of pulses.<br>The Isolation Subtraction method finds the PCC parameters record for the corresponding isolation pulse and subtracts the selected isolation pulse from the selected nominal pulse. | calCombinationMethodType | 1 |

### 6.3.4 Timeline Instrument Auxiliary Parameters

The timeline instrument auxiliary parameters describe the sequence of ISPs expected for each of the SAR image acquisition modes.

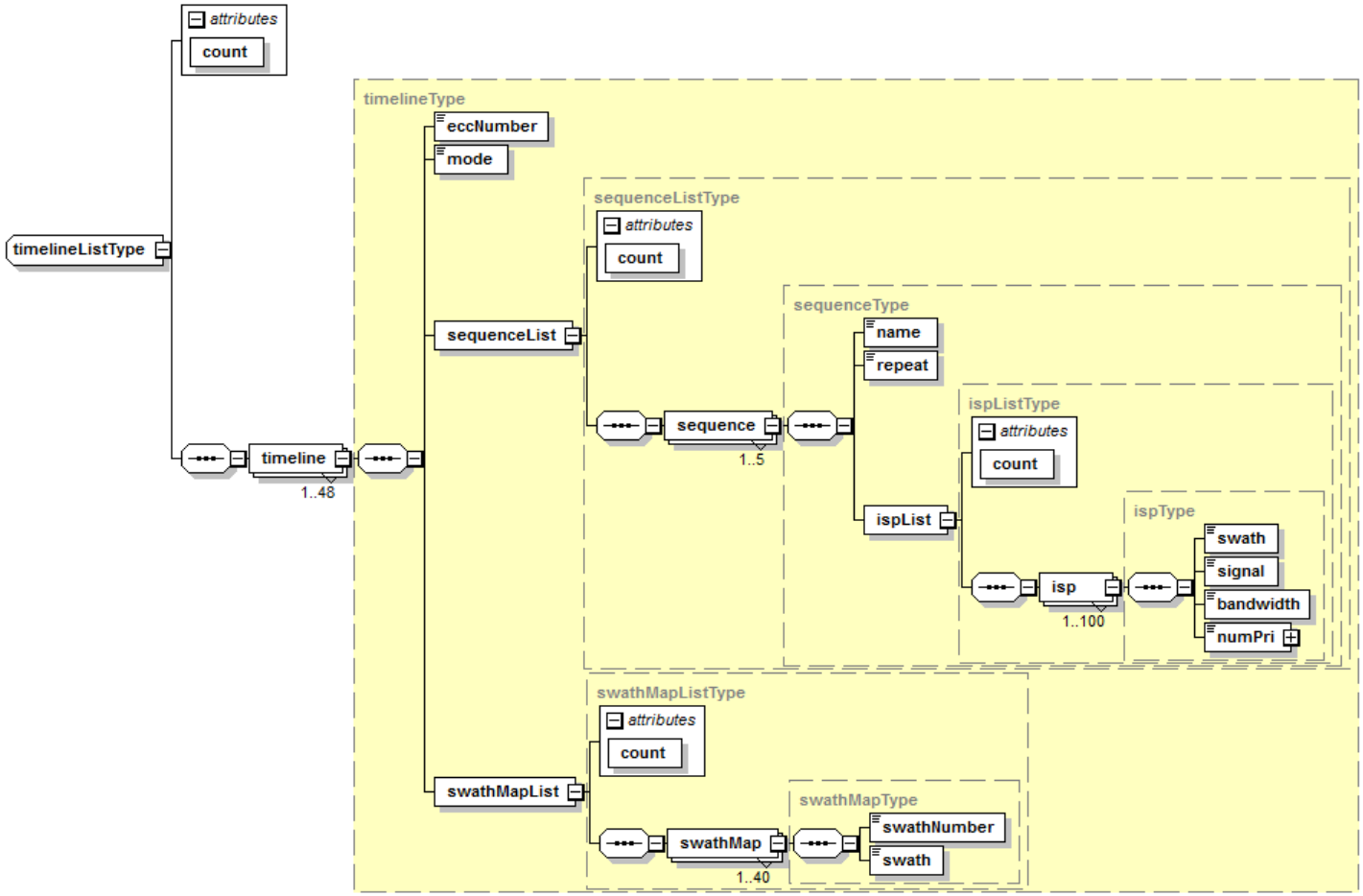


**Figure 6-5 Timeline Instrument Auxiliary Parameters**

**Table 6-18 Data Type - timelineListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of timeline entries in the list.* | *unsignedInt* | *required* |
| timeline | Each timeline element describes the expected packet transmission sequence for one of the operational modes of the satellite. The SPPDU [A-10] allows for up to 48 entries numbered from 0-47. At minimum, this list must include an entry for the instrument measurement mode of the data being processed. | timelineType | 1 .. 48 |

**Table 6-19 Data Type - timelineType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| eccNumber | Event Control Code (ECC) number. This field uniquely identifies the ECC program number for this instrument mode and is used by the IPF for timeline selection. | eccNumberType | 1 |
| mode | Instrument mode. This field identifies the instrument mode to which this timeline entry applies. | sensorModeType | 1 |
| sequenceList | Sequence list. This element is a list of activity sequences that together form the expected transmission sequence from the SAR instrument for the data take. | sequenceListType | 1 |
| swathMapList | Swath mapping list. This element is a list of the swaths applicable to this ECC program and provides a mapping between the swath number fields in the source packet headers and the logical instrument swath name to which they apply. | swathMapListType | 1 |

**Table 6-20 Data Type - sequenceListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of sequence entries in the list.* | *unsignedInt* | *required* |
| sequence | This record defines the sequence of expected ISPs for one distinct activity within the data take. The number of sequences is arbitrary but there are 5 slots nominally allocated for:<br>1. an initial noise measurement sequence;<br>2. an initial calibration sequence;<br>3. an image acquisition sequence;<br>4. a final calibration sequence; and,<br>5. a final noise measurement sequence. | sequenceType | 1 .. 5 |

**Table 6-21 Data Type - sequenceType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| name | The name of the activity within the data acquisition to which this sequence belongs. This field is not used by the IPF for processing. It is for informative purposes only and so the range of the field is unbounded. | string | 1 |
| repeat | Sequence repeat flag. For the imaging sequence, this field shall be set to “true” to indentify the ispList that represents the imaging operation. This field shall be set to “false” for all other sequences. | bool | 1 |
| ispList | ISP list. This element contains a list of the expected packets within this sequence in the order they should be received. The number of ISP entries is arbitrary but there are 30 slots defined to capture the worst case EW echo acquisition with PCC2 sequences at the end of each burst. | ispListType | 1 |

**Table 6-22 Data Type - ispListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *The number of ISP records within the list.* | *unsignedInt* | *required* |
| isp | The ISP element describes one unique, or a series of unique transmission packets. The packets are identified by the packet signal type and contain the number of PRIs expected for this packet type. | ispType | 1 .. 100 |

**Table 6-23 Data Type - ispType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swath | Canonical name of the swath used to acquire the packet(s). | swathType | 1 |
| signal | Signal type. | signalType | 1 |
| bandwidth | Signal bandwidth type. | bandwidthType | 1 |
| numPri | The number of packets of this signal type expected in series. | uint32 | 1 |

**Table 6-24 Data Type - swathMapListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of swathMap records in the list.* | *unsignedInt* | *required* |
| swathMap | Provides a mapping bewteen a particular swath number from the secondary source packet headers to a logical swath within the instrument mode. In theory the swath number can vary by signal type so the maximum number of swath map elements is 5 swaths * 8 signal types = 40 records. This is the worst case for EW. | swathMapType | 1 .. 40 |

**Table 6-25 Data Type – swathMapType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| swathNumber | The swath number from the source packet secondary header to map to a particular logical swath within the instrument mode. | swathNumberType | 1 |
| swath | The logical swath to which the swath number applies. | swathType | 1 |

### 6.3.5 Decoding Instrument Auxiliary Parameters

The decoding instrument auxiliary parameters contain the LUTs necessary for the IPF to perform FDBAQ/BAQ decoding on the ISPs.

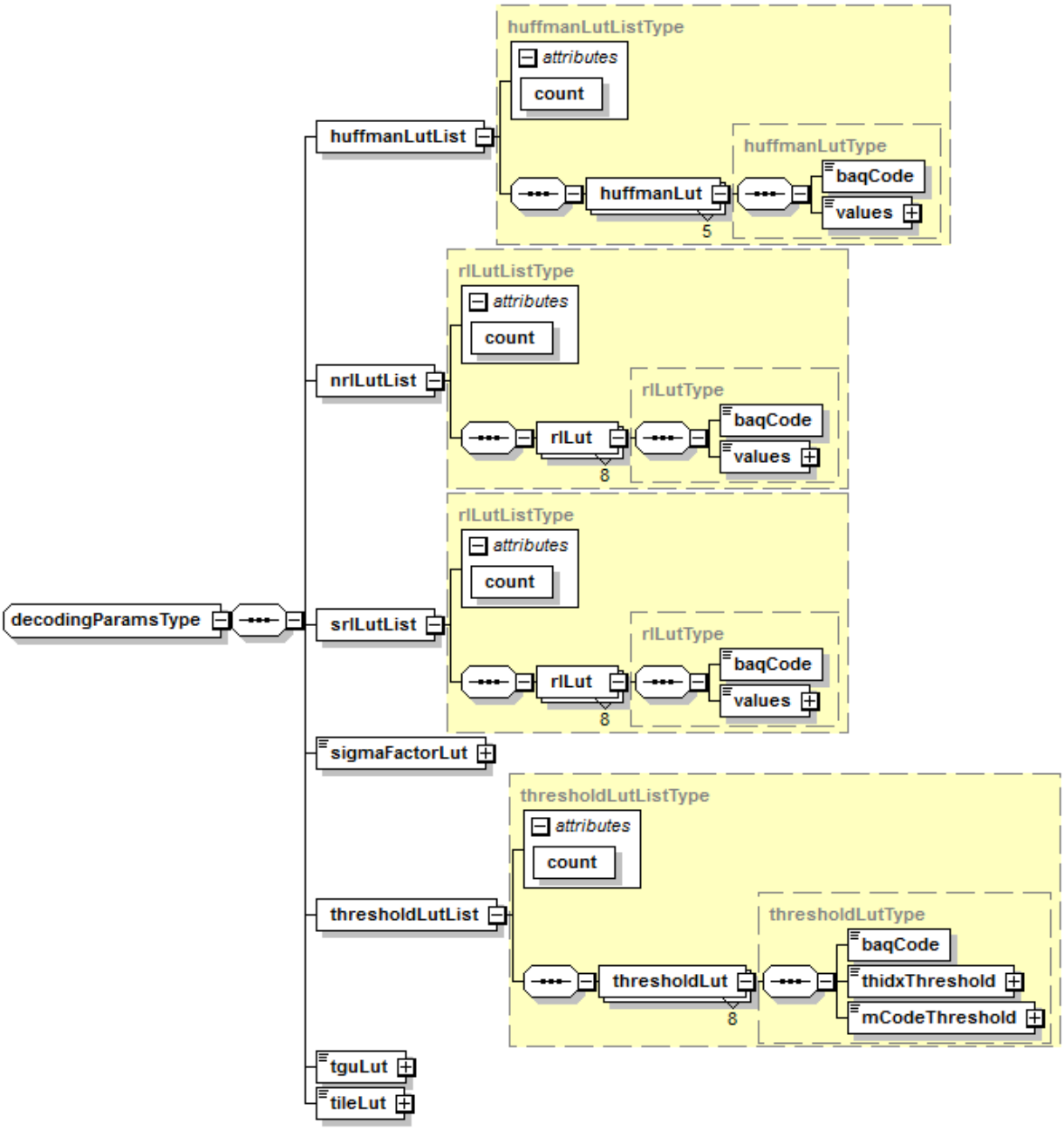


**Figure 6-6 Decoding Instrument Auxiliary Parameters**

**Table 6-26 Data Type - decodingParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| huffmanLutList | Huffman decoding LUT list. This element contains the Huffman decoding LUTs required to retrieve the HCode value from FDBAQ encoded user data. There is one LUT for each Bit Rate Code for a total of 5. | huffmanLutListType | 1 |
| nrlLutList | Normalised Reconstruction Levels LUT list. This element contains the NRL LUTs required to retrieve the normalised reconstructed sample values from the BAQ encoded data. There is one LUT per BAQ mode/FDBAQ Bit Rate Code for a total of 8 LUTs. | rlLutListType | 1 |
| srlLutList | Simple Reconstruction Parameters LUT list. This element contains the Simple Reconstruction Parameters LUTs required to retrieve the simple reconstructed sample values from the BAQ encoded data. There is one LUT per BAQ mode/FDBAQ Bit Rate Code for a total of 8 LUTs. | rlLutListType | 1 |
| sigmaFactorLut | Sigma Factors LUT. This LUT contains the values used to upscale the normalised reconstructed samples. The sigma factors in the table are indexed by the Threshold Index (THIDX) extracted from the BAQ block. This vector contains 255 single precision floating point numbers separated by spaces. | floatArray | 1 |
| thresholdLutList | Raw data decoding control LUT list. This element contains the parameters required to decode the BAQ and FDBAQ encoded data. | thresholdLutListType | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| tguLut | TGU temperature LUT. This LUT contains the values that are used to convert the TGU temperature codes extracted from the sub-commutated ancillary data in the ISP headers into the correct temperature value [degrees C]. The LUT contains a list of 128 single-precision floating values separated by spaces with the index of each entry, numbered 0 .. 127, corresponding to the code for which the temperature value applies. The index of each entry represents the code to convert, with the first entry in the list starting at index/code 0 and incrementing by 1 for each subsequent entry. For example, for the following definition:<br><tguLut count=”128”>116.14 115.02 113.9 . . .</tguLut><br>Represents the following:<br>Code 0 = 116.14 degC<br>Code 1 = 115.02 degC<br>Code 2 = 113.9 degC<br>etc… | floatArray | 1 |
| tileLut | Tile temperature LUT. This LUT contains the values that are used to convert the EFE and active TA temperature codes extracted from the sub-commutated ancillary data in the ISP headers into the correct temperature value [degrees C]. The LUT contains a list of 256 single-precision floating values separated by spaces with the index of each entry, numbered 0 .. 255, corresponding to the code for which the temperature value applies. The index of each entry represents the code to convert, with the first entry in the list starting at index/code 0 and incrementing by 1 for each subsequent entry. For example, for the following definition:<br><tileLut count=”256”>-51.38 -47.38 -44.38 . . .</tileLut><br>Represents the following:<br>Code 0 = -51.38 degC<br>Code 1 = -47.38 degC<br>Code 2 = -44.38 degC<br>etc… | floatArray | 1 |

**Table 6-27 Data Type - huffmanLutListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of Huffan decoding LUTs in the list.* | *unsignedInt* | *required* |
| huffmanLut | Huffman decoding LUT. This element contains the Huffman binary tree values for the applicable Bit Rate Code. The MCode is recovered by applying the values in the decoding LUT to the extracted HCode. | huffmanLutType | 5 |

**Table 6-28 Data Type - huffmanLutType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| baqCode | Bit Rate Code (as extracted from the BAQ block) to which this LUT applies. | baqCodeType | 1 |
| values | Huffman binary decoding tree values. The tree is implemented using a coded integer array in which starting at the root, the left side is defined and then the right side is defined using integer values as follows: Each node is identified by a 0 followed by a 0 or 1 representing the value of the node. Each leaf is identified by a 1 followed by a 0 or 1 representing the value of the leaf followed by the MCode value. | intArray | 1 |

**Table 6-29 Data Type - rlLutListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of Reconstruction Level LUTs in the list.* | *unsignedInt* | *required* |
| rlLut | Reconstruction Levels LUT. This LUT contains the RL values used to retrieve the reconstructed sample values from the BAQ encoded data. The RL in the table are indexed by:<br>1. the Bit Rate Code extracted from the BAQ data block for FDBAQ compression; and,<br>2. the BAQ mode (3-bit, 4-bit or 5-bit) for BAQ compression. | rlLutType | 8 |

**Table 6-30 Data Type - rlLutType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| baqCode | Index code for which the LUT applies. For FDBAQ compression this is the Bit Rate Code extracted from the BAQ block and for BAQ compression this is the BAQ mode. | baqCodeType | 1 |
| values | RL LUT values. This element contains up to sixteen (fifteen for NRL, sixteen for SRL) single precision floating point values separated by spaces, one entry for each MCode or THIDX value. Note that some MCodes/THIDX are not applicable for some FDBAQ and BAQ modes so in this case the entry shall be "NaN" to signify an invalid index. | floatArray | 1 |

**Table 6-31 Data Type - thresholdLutListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *Number of thresholdLuts in the list.* | *unsignedInt* | *required* |
| thresholdLut | Threshold LUT containing the thresholds needed to decode the BAQ and FDBAQ encoded data. There is one record for each of the BAQ-modes/FDBAQ-BRCs for a total of 8. | thresholdLutType | 8 |

**Table 6-32 Data Type - thresholdLutType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| baqCode | BAQ-mode/FDBAQ-BRC to which this set of thresholds applies. | baqCodeType | 1 |
| thidxThreshold | THIDX threshold used to determine whether to use the simple reconstruction method or the normal reconstruction method. If the THIDX extracted from the data is less than or equal to this threshold, then the simple reconstruction method is used; otherwise, the normal reconstruction method is used. | int32 | 1 |
| mCodeThreshold | Mcode threshold used in simple reconstruction to determine whether to use the extracted Mcode or the the simple reconstruction LUT. If the Mcode extracted from the data is less than this threshold, then the extracted Mcode is used; otherwise, the simple reconstruction LUT is used. | int32 | 1 |

# 7 L2 PROCESSOR PARAMETERS AUXILIARY DATA

## 7.1 Overview

The L2 processor parameters auxiliary file contains parameters required by the IPF for producing L2 products with the desired properties. It allows the fine-tuning of the processor and the selection of specific processing options.

This auxiliary product is expected to be updated infrequently and will change only when the characteristics of the output products need to be modified or when the IPF software is modified.

### 7.1.1 Product Structure

Table 4-1 describes the physical structure and data file components included in the L2 Processor Parameters Auxiliary Product as well as a nominal estimated size for each.

**Table 7-1 Sentinel-1 L2 Processor Parameters Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| MMM_AUX_PP2_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| mmm-aux-pp2.xml | Auxiliary Data | 1 | 150 KB |
| support/ | Directory | 1 | N/A |
| s1-aux-pp2.xsd | Representation Data | 1 | 13 KB |
| s1-object-types.xsd | Representation Data | 1 | 48 KB |

### 7.1.2 Naming Standard

Naming for the L2 processor parameters auxiliary product follows the standard defined in Section 3.2.1 and naming of the component files within the L2 processor parameters auxiliary product follows the standard defined in Section 3.2.2.

## 7.2 Data File Definition

The schema file for the definition of the L2 processor parameters auxiliary file is specified in Appendix A4.

## 7.3 Data File Content

Figure 7-1 presents a high-level graphical view of the L2 processor parameters auxiliary data contained in the auxiliary data file mmm-aux-pp2.xml, and a detailed tabular view of the L2 processor parameters auxiliary file is presented in the tables below.

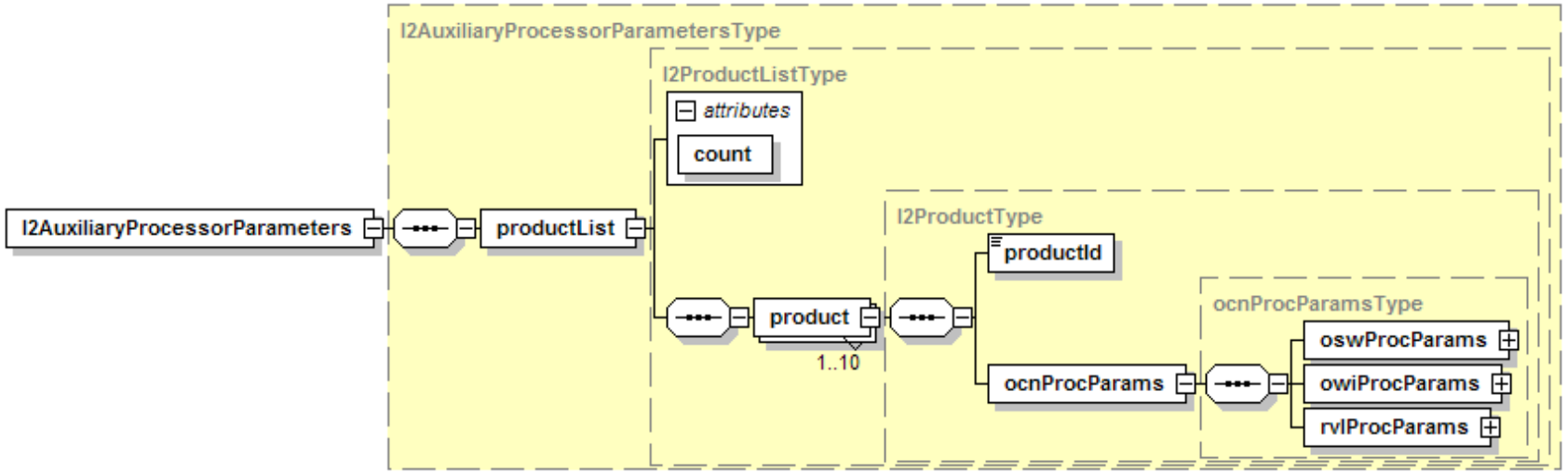


**Figure 7-1 L2 Auxiliary Processor Parameters**

**Table 7-2 Element - l2AuxiliaryProcessorParameters**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| productList | List of L2 standard products containing the applicable auxiliary parameters for each. This list contains an entry for each product the IPF is capable of generating, indexed by its unique product identifier. | l2ProductListType | 1 |

**Table 7-3 Data Type - l2ProductListType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *count* | *The number of elements contained in the list. There is an entry for each standard product type that the IPF is capable of generating.* | *unsignedInt* | *required* |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| product | Product auxiliary parameters. This DSR contains all of the auxiliary parameters required to process a single product. The parameters are stored in structures that are grouped together by the logical processing steps used during image creation. This DSR contains the productId element which is used to identify the product that this set of parameters applies to. The parameters within this DSR and its children are not polarisation dependent and in general apply to the entire product; however, in some cases a distinction must be made amongst swaths and when this is necessary the records are indexed with a swath identifier. | l2ProductType | 1 .. 10 |

**Table 7-4 Data Type - l2ProductType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| productId | L2 Product type identifier to which this set of parameters applies. The productId is used to index and find the correct set of auxiliary parameters for each product the IPF is capable of generating. This field corresponds to the first 9 characters of the product type identifiers listed in the Job Order File_Type field. For example, the S1 IPF ICD [A-5] defines a product identifier for SM OCN standard products as “SM_OCN__1S”, so the parameters that correspond to this product are identified by the string “SM_OCN__1”. | productIdType | 1 |
| ocnProcParams | OCN processing auxiliary parameters. This record contains the auxiliary parameters required during OSW, OWI, and RVL processing. | ocnProcParamsType | 1 |

**Table 7-5 Data Type - ocnProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| oswProcParams | OSW component processor parameters. | oswProcParamsType | 1 |
| owiProcParams | OWI component processor parameters. | owiProcParamsType | 1 |
| rvlProcParams | RVL component processor parameters. | rvlProcParamsType | 1 |

### 7.3.1 OSW Processing Parameters

The OSW Processing Parameters section contains the parameters that control and fine tune the OSW component processing of the IPF. Figure 7-2 presents a high-level graphical overview of the OSW processing parameters and the subsequent tables describe the content of this DSR.

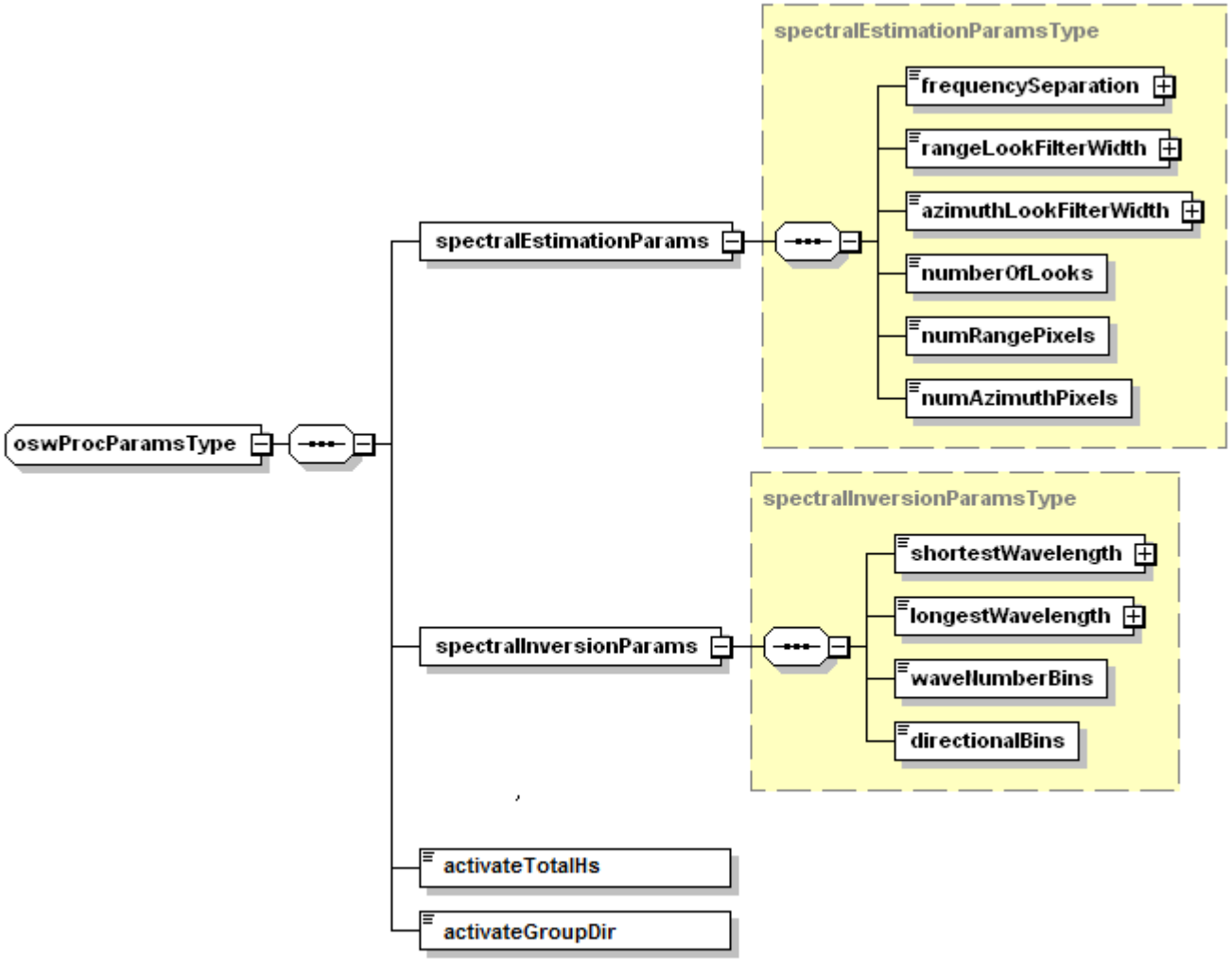
spectralEstimationParamsType
frequencySeparation
rangeLookFilterWidth
azimuthLookFilterWidth
numberOfLooks
numRangePixels
numAzimuthPixels
spectralEstimationParams
oswProcParamsType
spectralInversionParamsType
shortestWavelength
longestWavelength
waveNumberBins
directionalBins
spectralInversionParams
activateTotalHs
activateGroupDir

**Figure 7-2 OSW Processor Parameters**

**Table 7-6 Data Type - oswProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| spectralEstimationParams | This record contains the auxiliary parameters required for OSW spectral estimation. | spectralEstimationParamsType | 1 |
| spectralInversionParams | This record contains the auxiliary parameters required for OSW spectral inversion. | spectralInversionParamsType | 1 |
| activateTotalHs | Introduced with IPF 2.90 but not considered before IPF 3.50.<br>Allow activating the computation of the total significant wave height, value=True to activate the calculation, otherwise False<br>Optional attribute named "beam" to allow decoupling the activation depending on the beam for WV ( ie: allowed attribute value "WV1" and "WV2") | bool | 1, 2 |
| activateGroupDir | Introduced with IPF 2.90. It will be used by future IPF versions.<br>Allow activating the computation of the peak direction from Group analysis, value=True to activate the calculation, otherwise False | bool | 1 |
| activateNoiseCorrection | Introduced with IPF version IPF 3.50<br>Allow to trigger the denoising on OSW | bool | 1 |

**Table 7-7 Data Type - spectralEstimationParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| frequencySeparation | Frequency separation of neighbouring looks [Hz]. | double | 1 |
| rangeLookFilterWidth | Range look filter width [Hz]. | double | 1 |
| azimuthLookFilterWidth | Azimuth look filter width [Hz]. | double | 1 |
| numberOfLooks | Number of individual looks. | unsignedInt | 1 |

| | | | |
|---|---|---|---|
| numRangePixels | Number of range pixels from input L1 product to be used in the estimation of the OSW [pixels]. Only used for SM, for WV all range pixels are used. | unsignedInt | 1 |
| numAzimuthPixels | Number of azimuth pixels from input L1 product to be used in the estimation of the OSW [pixels]. Only used for SM, for WV all azimuth pixels are used. | unsignedInt | 1 |

**Table 7-8 Data Type - spectralInversionParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| shortestWavelength | Shortest wavelength of output polar grid [m]. | double | 1 |
| longestWavelength | Longest wavelength of output polar grid [m]. | double | 1 |
| waveNumberBins | Number of wavenumber bins in polar grid. | unsignedInt | 1 |
| directionalBins | Number of directional bins in polar grid. | unsignedInt | 1 |
| velthresh | Introduced for IPF version > IPF3.50<br>Parameters relative to velocity bunching<br>The parameter has an attribute named "beam" to allow decoupling the activation depending on the beam for WV ( ie: allowed attribute value WV1 ,WV2, S1,S2,S3, S4,S5,S6) | double | 1, 2, 6 |

### 7.3.2 OWI Processing Parameters

The OWI Processing Parameters section contains the parameters that control and fine tune the OWI component processing of the IPF. Figure 7-3 presents a high-level graphical overview of the OWI processing parameters and the subsequent tables describe the content of this DSR.

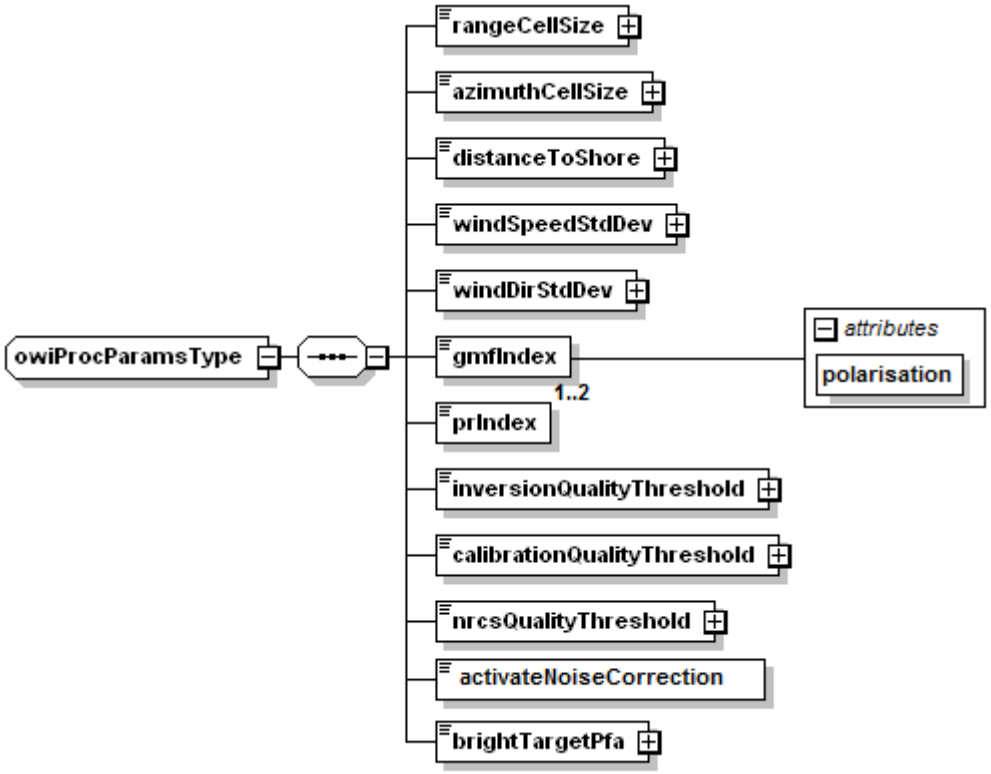


**Figure 7-3 OWI Processor Parameters**

**Table 7-9 Data Type - owiProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rangeCellSize | Size of the SAR derived wind field in the range direction [m]. Wind cells should typically be square therefore nominally rangeCellSize = azimuthCellSize. | double | 1 |
| azimuthCellSize | Size of the SAR derived wind field in the azimuth direction [m]. Wind cells should typically be square therefore nominally azimuthCellSize = rangeCellSize. | double | 1 |
| distanceToShore | Distance to shore where the processing is not performed [Km]. | double | 1 |
| windSpeedStdDev | Standard deviation error of the wind speed provided by ancillary wind information [m/s]. | double | 1 |
| windDirStdDev | Standard deviation error of the wind direction provided by ancillary wind information [degrees]. | double | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| gmfIndex | Index or name of the Geophysical Model Function (GMF) to be used for the wind inversion, used for either for VV or HH polarisation data. Options supported are integers in the range [ 0, 21]<br>0=gmf_cmod2_i3_nn, 1= gmf_cmod_ifr2_nn, 2=gmf_cmod_ifr2_bc_nn,<br>3= gmf_cmod4_nn, 4=gmf_cmod2_i3, 5=gmf_cmod_ifr2, 6=gmf_cmod_ifr2_bc, 7=gmf_cmod4, 8=gmf_cmod5, 9=gmf_cmodcross2pol, 10=gmf_cmodrs2cross4pol, 11=gmf_cmodrs2cross4pol_vachon, 12=gmf_cmod5n, 13=gmf_cmod5h, 14=gmf_cmod5na, 15=gmf_cmod6, 16=gmf_sarmodVV, 17=gmf_sarmodHH, 18=gmf_cmodcross_h14e, 19=gmf_cmodcross_h14s, 20=gmf_cmodcross_ms1a, 21=gmf_cmodrs2cross2pol<br>The concerned polarisation is discriminated by the attribute named "polarisation" with supported value 'VV' or 'HH' | Integer or string | 2 |
| prIndex | Index for the Polarisation Ratio Function to be used for the wind inversion (used for HH polarisation data). This function converts HH NRCS into VV NRCS before applying the GMF specified by gmfIndex (2nd value) to retrieve the wind. Options supported are integers from 0 to 3.<br>0= Thomson[1] ; 1= Mouche-1[2], 2= Elfouhaily[3]; ; 3= Mouche2 [3]. | integer | 1 |
| inversionQualityThreshold | Value above which minimization in the inversion is considered low quality. Valid value range is between 0 and 1e29. | double | 1 |
| calibrationQualityThreshold | Value above which the calibration of the product is considered to be incorrect [dB]. Valid value range is between 0 and 10 dB. | double | 1 |

[1] Thompson D.R, T. Elfouhaily, B. Chapron , "Polarization ratio for Microwave backscattering from the Ocean Surface at low to moderate incidence angles," *Proc. Int. Geoscience and Rem. Sensing Symp*., Seattle WA, 1998

[2] Alexis Mouche, D. Hauser,V. Kudryavtsev and JF. Daloze, "Multi polarization ocean radar cross-section from ENVISAT ASAR observations, airborne polarimetric radar measurements and empirical or semi-empirical models", *ESA ERS/ENVISAT Symposium*, Salzburg, September 2004

[3] T. Elfouhaily *Modele couple vent/vagues et son application a la teledetection par micro-ondes de la surface de la mer*, PhD thesis, Universite de Paris 7, 20 Novembre 1996

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| nrcsQualityThreshold | Value above which the NRCS estimated at the SAR wind cell resolution is considered as low quality [dB]. Valid value range is between -30 and 10 dB. | double | 1 |
| brightTargetPfa | Probability of false alarm for the removal of bright target. | double | 1 |
| activateNoiseCorrection | THIS ELEMENT IS NOT USED BY IPF 2.70.<br>IT WILL BE USED FOR FUTURE VERSIONS<br>Allow activating the noise correction, value=True to activate the noise correction, otherwise False | bool | 1 |

### 7.3.3 RVL Processing Parameters

The RVL Processing Parameters section contains the parameters that control and fine tune the RVL component processing of the IPF. Figure 7-4 presents a high-level graphical overview of the RVL processing parameters and the subsequent table describes the content of this DSR.

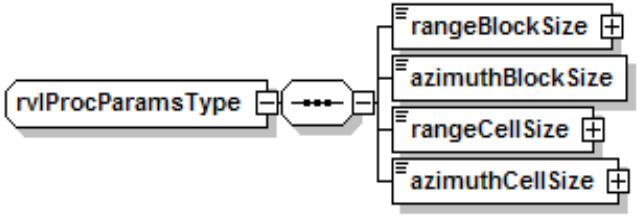


**Figure 7-4 RVL Processor Parameters**

**Table 7-10 Data Type - rvlProcParamsType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| rangeBlockSize | Size of the Doppler estimation block in the range direction [m]. | double | 1 |
| azimuthBlockSize | Size of the Doppler estimation block in the azimuth direction [m]. | double | 1 |
| rangeCellSize | Size of grid cell interval in range direction [m]. | Double | 1 |
| azimuthCellSize | Size of grid cell interval n azimuth direction [m]. | Double | 1 |

# 8 SIMULATED CROSS SPECTRA AUXILIARY DATA

## 8.1 Overview

The OSW processing algorithm requires a look-up table to predict the modulation transfer function (MTF) and to remove non-linear effects from the cross spectra. There is one look-up table for each swath and polarisation (HH and VV only). The basic content of the look-up table is the simulated cross-spectrum for a given ocean swell spectrum, computed from the input wind speed, direction and inverse wave age, and an estimate of the MTF for the given wind field:

- Simulated Cross-Spectra are computed for a wide range of wind speeds and directions. A parameterization of the cross-spectra is stored in the table.
- The real aperture MTF amplitude;
- The wind speed range array;
- The wind direction array; and,
- The inverse wave age array.

This auxiliary product will change only when the OSW algorithm changes or when the cross spectra simulation model changes.

### 8.1.1 Product Structure

Table 8-1 describes the physical structure and data file components included in the calibration auxiliary product as well as an estimated size for each.

**Table 8-1 Sentinel-1 Simulated Cross Spectra Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| MMM_AUX_SCS_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| s1--aux-scs-sss-pp.info | Auxiliary Data (information) | 1 .. 32 | 133 KB |
| s1--aux-scs-sss-pp.nlin | Auxiliary Data (simulated spectra) | 1 .. 32 | 83 MB |
| support/ | Directory | 0 .. 1 | N/A |
| s1-object-types.xsd | Representation Data | 0 .. ∞ | N/A |

### 8.1.2 Naming Standard

Naming for the Simulated Cross Spectra auxiliary product follows the standard defined in Section 3.2.1 and naming of the component files within the Simulated Cross Spectra auxiliary product follows the standard defined in Section 3.2.2.

## 8.2 Data File Definition

The Simulated Cross Spectra auxiliary product contains data files that follow the format described in Section 8.3.

## 8.3 Data File Content

The Simulated Cross Spectra auxiliary product contains two data files per swath for each of the SM and WV swaths, and two sets of files per swath for each co-polarisation (HH and VV):

- The information file (.info) which is an ASCII file containing general information about the simulated spectra, as well as a look-up table for indexing into the simulated spectra.
- The data file (.nlin) which contains a sequence of two-dimensional spectra, which can be indexed and extracted using information from the information file.

Each Simulated Cross Spectra auxiliary product will contain all of the modes and swaths for a mission. For Sentinel-1, it will contain both S1A and S1B.

Table 8-2 describes the description part of the information file. This is followed by a LUT, where each row of the LUT contains values for tau, U10, dir, iwa, dSigma, sigma0 and lambda.

**Table 8-2 Simulated Cross Spectra Information File**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| PRODUCT | Name of the simulated spectra product | STR | 1 |
| XSIZE | X-dimension of each simulated 2D cross spectra in the data file | LON | 1 |
| YSIZE | Y-dimension of each simulated 2D cross spectra in the data file | LON | 1 |
| NCHANNELS | The number of simulated spectra in the file. | LON | 1 |
| DATATYPE | The data type by the IDL-language data type code (DATATYPE=6 represents 8-byte complex float) | LON | 1 |
| ENDIAN | “LITTLE” or “BIG” | STR | 1 |
| XBINSIZE | Range direction bin size of the simulated spectrum (rad/m) | FLT | 1 |
| YBINSIZE | Azimuth direction bin size of the simulated spectrum (rad/m) | FLT | 1 |
| INCANGLE | Reference incidence angle of the swath to which the simulated spectra in this file apply (degrees) | FLT | 1 |
| RANGEDIST | Slant range distance (m). | FLT | 1 |
| RADARVEL | Radar platform velocity (m/s). | FLT | 1 |
| WAVENUMBER | Radar wavenumber (rad/m). | FLT | 1 |
| POLARIZATION | Polarisation of the simulated cross spectra, either “HH” or “VV”. | STR | 1 |
| TAU*LAYOUT | The layout of the TAU column in the look-up table, either “LINEAR” or “LOGARITHMIC”. | STR | 1 |
| TAU*NVALUES | The number of values per simulated cross spectra in the TAU column of the look-up table | STR | 1 |
| TAU*MIN | The minimum value per simulated cross spectra in the TAU column of the look-up table | STR | 1 |
| TAU*MAX | The minimum value per simulated cross spectra in the TAU column of the look-up table | STR | 1 |
| U10*LAYOUT | The layout of the U10 column in the look-up table, either “LINEAR” or “LOGARITHMIC”. | STR | 1 |
| U10*NVALUES | The number of values per simulated cross spectra in the U10 column of the look-up table | STR | 1 |
| U10*MIN | The minimum value per simulated cross spectra in the U10 column of the look-up table | STR | 1 |
| U10*MAX | The minimum value per simulated cross spectra in the U10 column of the look-up table | STR | 1 |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| DIR*LAYOUT | The layout of the DIR column in the look-up table, either “LINEAR” or “LOGARITHMIC”. | STR | 1 |
| DIR*NVALUES | The number of values per simulated cross spectra in the DIR column of the look-up table | STR | 1 |
| DIR*MIN | The minimum value per simulated cross spectra in the DIR column of the look-up table | STR | 1 |
| DIR*MAX | The minimum value per simulated cross spectra in the DIR column of the look-up table | STR | 1 |
| IWA*LAYOUT | The layout of the IWA column in the look-up table, either “LINEAR” or “LOGARITHMIC”. | STR | 1 |
| IWA*NVALUES | The number of values per simulated cross spectra in the IWA column of the look-up table | STR | 1 |
| IWA*MIN | The minimum value per simulated cross spectra in the IWA column of the look-up table | STR | 1 |
| IWA*MAX | The minimum value per simulated cross spectra in the IWA column of the look-up table | STR | 1 |
| PROC*FACILITY | Facility where the simulated cross spectra files were generated. | STR | 1 |
| PROC*SOFTWARE | Software used to generate the simulated cross spectra. | STR | 1 |
| PROC*VERSION | Version of the software used to generate the simulated cross spectra. | STR | 1 |
| PROC*DATE | Date when the simulate cross spectra files were created. | STR | 1 |
| MTFWV | Factor to apply to the MTF in order to adjust the overall effective Hs performances.<br>Factor introduced in IPF version 3.3.0 | STR | 1 |

The data file contains two dimensions binary spectra The size of the stored spectral arrays are given by the XSIZE and YSIZE tags. DATATYPE describes the data type by the IDL-language data-type code (DATATYPE}=6 represent 8-byte complex float) and ENDIAN the computer architecture dependent unformatted number representation (ENDIAN can be either “BIG” or “LITTLE”'). The range and azimuth binsize of the spectra, are given by the XBINSIZE and YBINSIZE tags. Only the upper half-plane of the spectra are stored representing both positive and negative range wave-numbers but only positive azimuth wave-numbers.

The tag-structures TAU, U10, DIR and IWA each contain the following sub-tags: LAYOUT describes if the parameter is sampled linearly or logarithmic, NVALUES the total number of samples and MIN and MAX the minimum and maximum value of the parameter.

The spectra in the data-file and the information about them in the parameter part of the info-file are written in the same order, from indexing fastest (inner-loop) through

and to indexing slowest (outer-loop). The total number of channels are therefore given by:

NCHANNELS =
(TAU*NVALUES)(U10*NVALUES)(DIR*NVALUES)(IWA*NVALUES)

# 9 ORBIT AUXILIARY DATA

## 9.1 Overview

Orbit auxiliary products contain information about the position of the satellite during the acquisition of SAR data. Orbit information may be predicted or generated by the Precise Orbit Determination (POD) service. There are three types of orbit auxiliary data:

- Restituted Orbit (AUX_RES) containing orbit state vectors (OSV) based on the orbit determination performed by the POD service and used for near-real-time processing; and,
- Precise Orbit Ephemeris (AUX_POE) containing orbit state vectors based on the orbit determination performed by the POD service and used for offline processing. The AUX_POE contains the most precise set of orbit state vectors.
- Predicted Orbit (AUX_PRE) containing orbit state vectors based on the orbit prediction performed by the POD service and used for NRT processing. Tthe AUX_PREORB contains more precise set of orbit state vectors than AUX_RES, with similar performances to the one of AUX_POE

The frequency of update for the orbit auxiliary data is TBD by ESA (as per [A-3]).

AUX_PREORB files are only supported started by IPF version 3.2.0.

### 9.1.1 Product Structure

The structure of the orbit auxiliary data is described in [A-3].

### 9.1.2 Naming Standard

The naming standard of the orbit auxiliary data follows the standard defined in [A-3].

### 9.1.3 Data File Definition

The definition of the orbit auxiliary data is specified in [A-3].

### 9.1.4 Data File Content

The content of the orbit auxiliary data is specified in [A-3].

# 10 ATTITUDE AUXILIARY DATA

## 10.1 Overview

Attitude auxiliary products contain information about the pointing of the satellite during the acquisition of SAR data. The attitude auxiliary data contains the restituted attitude data computed by the Precise Orbit Determination (POD) service based on input data from the Sentinel-1 PDGS.

The frequency of update for the attitude auxiliary data is TBD by ESA (as per [A-3]).

### 10.1.1 Product Structure

The structure of the attitude auxiliary data is described in [A-3].

### 10.1.2 Naming Standard

The naming standard of the attitude auxiliary product follows the standard defined in [A-3].

## 10.2 Data File Definition

The definition of the attitude auxiliary data is specified in [A-3].

## 10.3 Data File Content

The content of the attitude auxiliary data is specified in [A-3].

# 11 ECMWF ATMOSPHERIC MODEL AUXILIARY DATA

## 11.1 Overview

The ECMWF atmospheric model provides a forecast of the wind speed and direction at 10 m above the sea surface with spatial and temporal resolution of at least 0.25 degrees. Finer resolution is supported as well (for instance 0.1 degrees).

This is data used for OWI, OSW and RVL processing.

The ECMWF data files contain many parameters, only some of which are used for Sentinel-1 product processing. In reference to [A-6], the parameters used for Sentinel-1 are:

- 165, 10 metre U (m/s)
- 166, 10 metre V (m/s)

The ECMWF atmospheric model auxiliary data must be updated with a new forecast required at 3-hour intervals (or lower) from 3 to 144 hours, based on 12 UTC and 00 UTC data. Smaller intervals allow to capture fast changing atmospheric conditions.

Each AUX_WND product corresponds to the forecast of wind situation (see above 10 meters U and V components in m/s) at a given date:

- The validity date of the product corresponds to the date of the forecasted atmospheric situation
- The generation date of the corresponds to the generation date of the product itself. By design this date is after the one of the runs used for the forecast.

### 11.1.1 Product Structure

The atmospheric model from the ECMWF is in GRIB format and is wrapped in SAFE before being provided to the IPF. Table 11-1 describes the physical structure and data file components included in the ECMWF atmospheric model auxiliary product as well as an estimated size for each.

**Table 11-1 Sentinel-1 ECMWF Atmospheric Model Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Approx. Size |
|---|---|---|---|
| S1__AUX_WND_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |

| | | | |
|---|---|---|---|
| data/ | Directory | 1 | N/A |
| ccsmmddhhiimmddhhiiE.grb (as per [A-6]) | Auxiliary Data | 1 | 12 MB |
| support/ | Directory | 0 .. 1 | N/A |
| Optional support files. | Representation Data | 0 .. ∞ | N/A |

### 11.1.2 Naming Standard

Naming for the ECMWF atmospheric model auxiliary product follows the standard defined in Section 3.2.1 and naming of the component file within the ECMWF atmospheric model auxiliary product follows the standard defined in [A-6].

## 11.2 Data File Definition

The definition of the ECMWF atmospheric model auxiliary data is specified in [A-6].

## 11.3 Data File Content

The content of the ECMWF atmospheric model auxiliary data is specified in [A-6].

# 12 WAVEWATCH III MODEL AUXILIARY DATA

## 12.1 Overview

The stokes drift file from the WAVEWATCH III model is used to interpret the RVL component of the OCN product for all imaging modes.

The stokes drift is a vector calculated as the third order moment of the wave spectra. This cannot be estimated from SAR wave spectra since the stokes drift is mostly dependent on wind sea which is not imaged by SAR. Therefore this information has to come from a model that has a good physical representation of the shorter waves. Stokes drift forecast files in NetCDF format generated by the operational WAVEWATCH III model run at IFREMER are used.

The WAVEWATCH III model provides surface stokes drift velocity and direction for frequency range $f_{min}$=0.032 to $f_{max}$=0.72 Hz with spatial and temporal resolution of at least 0.5 degrees.

The WAVEWATCH III model stokes drift auxiliary data is updated with a new forecast every 6 hours.

Each AUX_WAV product corresponds to the forecast of swell situation at a given date:

- The validity date of the product corresponds to the date of the forecasted stokes drift
- The generation date of the product corresponds to the generation date of the product itself. By design this date is after the one of the runs used for the forecast.

### 12.1.1 Product Structure

The stokes drift data from the WAVEWATCH III model is in NetCDF format and is wrapped in SAFE before being provided to the IPF. Table 12-1 describes the physical structure and data file components included in the WAVEWATCH III Stokes Drift auxiliary product as well as an estimated size for each.

**Table 12-1 Sentinel-1 WAVEWATCH III Stokes Drift Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Est. Size |
|---|---|---|---|
| S1__AUX_WAV_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| ifremer-ww3-glob-30min-yyyymmddthhz*-uss.nc | Auxiliary Data | 1 | 1.4 MB |
| support/ | Directory | 0 .. 1 | N/A |
| Optional support files. | Representation Data | 0 .. ∞ | N/A |

## 12.1.2 Naming Standard

The naming standard of the WAVEWATCH III auxiliary product follows the standard defined in Section 3.2.1 and naming of the component file within the WAVEWATCH III model stokes drift auxiliary product follows the standard defined in Table 12-2 below:

**Table 12-2 WAVEWATCH III Model Stokes Drift File Naming Convention**

| File Name Element | Description |
|---|---|
| ifremer-ww3-glob-30min- | String literal “ifremer-ww3-glob-30min-”. The prefix “ifremer” may be replaced with any organisation name or identifier, for example: “shom”. |
| yyyy | Forecast produced from winds taken from the year yyyy. |
| mm | Forecast produced from winds taken from the month mm. |
| dd | Forecast produced from winds taken from the day dd. |
| t | String literal “t” |
| hh | Forecast produced from winds taken from the cycle hh. |
| z | String literal “z” (Zulu/UTC time). |
| * | Optional string literal corresponding to any character.<br>This can be used to specify the generation date of the netcdf file. |
| -uss | String literal “-uss” denoting the stokes drift data. |
| .nc | NetCDF file extension. |

## 12.2 Data File Definition

The definition of the WAVEWATCH III model stokes drift auxiliary data is specified in [A-7].

## 12.3 Data File Content

The content of the WAVEWATCH III model stokes drift auxiliary data is specified in [A-7]

# 13 SEA ICE AUXILIARY DATA

## 13.1 Overview

Sea ice edge data is used by the OWI algorithm. Wind inversion processing is not performed if sea ice concentration is greater than 10% in the imagette considered. Sea ice and open water areas are given in the product delivered by the Ocean and Sea Ice Satellite Application Facility (OSI SAF) at a spatial resolution of 10 km.

The sea ice auxiliary data is updated daily with new ice edge analysis. Note however that there is a lag of one day between when the sea ice is measured and when the sea ice auxiliary data is available.

Each AUX_ICE product corresponds to the forecast of swell situation at a given date:

- The validity date of the product corresponds to the date of the analysed sea ice situation (usually the day before the generation date)
- The generation date of the product corresponds to the generation date of the product itself. By design this date is after the one the analysis.

### 13.1.1 Product Structure

The sea ice data is in NetCDF format and is wrapped in SAFE before being provided to the IPF. Table 13-1 describes the physical structure and data file components included in the sea ice auxiliary product as well as an estimated size for each.

**Table 13-1 Sentinel-1 Sea Ice Auxiliary Product**

| File/Folder Name | File Type | Inclusion Criteria | Est. Size |
|---|---|---|---|
| S1__AUX_ICE_VYYYYMMDDTHHMMSS_GYYYYMMDDTHHMMSS.SAFE | | | |
| manifest.safe | Manifest | 1 | 5 KB |
| data/ | Directory | 1 | N/A |
| ice-edge-nh-yyyymmddhhmm.nc (as per [A-8]) | Auxiliary Data | 1 | 9.8 MB |
| ice-edge-sh-yyyymmddhhmm.nc (as per [A-8]) | Auxiliary Data | 1 | 7.6 MB |
| support/ | Directory | 0 .. 1 | N/A |

| | | | |
|---|---|---|---|
| Optional support files. | Representation Data | 0 .. ∞ | N/A |

### 13.1.2 Naming Standard

The naming standard of the sea ice auxiliary product follows the standard defined in Section 3.2.1 and naming of the component file within the sea ice auxiliary product follows the standard defined in [A-8]

## 13.2 Data File Definition

The definition of the sea ice auxiliary data is specified in [A-8].

## 13.3 Data File Content

The content of the sea ice auxiliary data is specified in [A-8].

# 14 EXCITATION COEFFICIENTS ERROR MATRIX AUXILIARY DATA

## 14.1 Overview

The Excitation Coefficients Error Matrix is a product derived from the Sentinel-1 RF Calibration (RFC) mode, which is a specific calibration mode meant to assess the instrument health and stability.

In particular, the RFC mode verifies the transmit (Tx) and receive (Rx) excitation coefficients to ensure the validity of the radiation beam patterns generated by the instrument. This is done through the error matrix product, which identifies changes in the excitation coefficients with respect to reference values (e.g. pre-launch reference).

The Error Matrix product is used by the RVL algorithm to derive the accurate Doppler estimation by synthesising a radiation beam pattern using an antenna model. Each Error Matrix product corresponds to a single polarisation, either H or V, and contains four components:

- EM_Txp
- EM_Rxp
- EM_EFERxp
- EM_TaRxp

For the purpose of antenna modelling in RVL, only the EM_Txh and EM_Rxh components are used.

The Error Matrix products will be updated over time in order to compensate for changes as a result of antenna component drifts or aging.

### 14.1.1 Product Structure

The structure of the excitation coefficients error auxiliary matrix data is described in [A-9].

### 14.1.2 Naming Standard

The naming standard of the excitation coefficients error matrix auxiliary product follows the standard is defined in [A-9].

## 14.2 Data File Definition

The definition of the excitation coefficients error matrix auxiliary data is specified in [A-9].

## 14.3 Data File Content

The content of the excitation coefficients error matrix auxiliary data is specified in [A-9].

# A SCHEMA DEFINITIONS

## A1 L1 Processor Parameters Auxiliary File

See attached file “s1-aux-pp1.xsd”

## A2 Calibration Auxiliary File

See attached file “s1-aux-cal.xsd”

## A3 Instrument Auxiliary File

See attached file “s1-aux-ins.xsd”

## A4 L2 Processor Parameters Auxiliary File

See attached file “s1-aux-pp2.xsd”

## A5 Primitive Data Types

See attached file “s1-object-types.xsd”

## A6 Sentinel-1 SAFE Specialisation Schemas

See attached schema files in the “xsd/” folder

# B MANIFEST FILE

The manifest file is documented in detail in the SAFE core specification [A-2]. The objective of this section is to define the mandatory components of the manifest file that will be present in all Sentinel-1 auxiliary products defined within this specification.

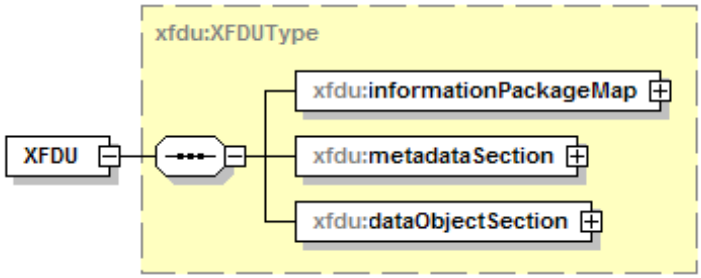


**Figure B-1 Sentinel-1 Manifest File Structure**

**Table B-1 Element - XFDU**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *version* | *The version attribute describes the location within the SAFE schema directory structure where the validating XFDU schema file for this manifest file resides.* | *string* | *required* |
| informationPackageMap | The information package map contains a high-level textual description of the product and references to all of the auxiliary data file components contained within the product. | informationPackageMapType | 1 |
| metadataSection | The metadata section contains a minimal set of wrapped product metadata that can be used for auxiliary product identification. | metadataSectionType | 1 |
| dataObjectSection | The dataOjectSection contains the dataObjects that represent the auxiliary data file components included in the auxiliary product. Each dataObject within the dataObjectSection represents a physical auxiliary data file on a local or external file system. | dataObjectSectionType | 1 |

# B1 Information Package Map

The manifest file contains exactly one information package map with one content unit as a child element. Beneath the first content unit is a list of content units that define the metadata and data objects within the product; that is, the components that make up the auxiliary product.

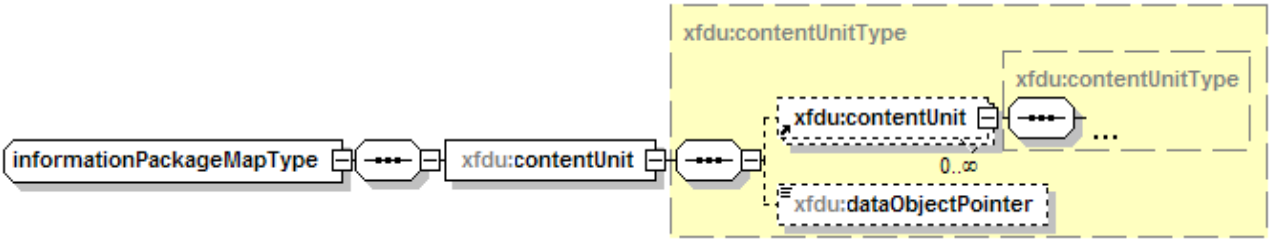


**Figure B-2 Information Package Map Structure**

**Table B-2 Data Type - informationPackageMapType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| contentUnit | The informationPackageMap contains exactly 1 contentUnit and this contentUnit catalogues the physical data components included in the auxiliary product. | contentUnitType | 1 |

**Table B-3 Data Type - contentUnitType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *ID* | *Unique identifier for this contentUnit.* | *ID* | *optional* |
| *unitType* | *Describes the type of data referenced by this content unit.* | *string* | *required* |
| *textInfo* | *A brief textual description of the information or data referenced by this content unit.* | *string* | *optional* |
| *repID* | *Identifier of the representation data set(s) applicable to this content unit. This can be a single item or a list with each item separated by a space.* | *IDREFS* | *optional* |
| *dmdID* | *Identifier of the metadata or annotation data set(s) applicable to this content unit. This can be a single item or a list with each item separated by a space.* | *IDREFS* | *optional* |
| *pdiID* | *Identifier of the preservation description information applicable to this content unit. For Sentinel-1 products this attribute shall always point to the "processing" wrapped metadata object.* | *IDREFS* | *optional* |

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| contentUnit | Content unit elements may include other content units or may be internal pointers to elements in the data object section. Content units are used to associate data objects with one or more metadata objects and present a view of these data/metadata associations. | contentUnitType | 0 .. ∞ |
| dataObjectPointer | Through the use of its dataObjectID attribute, this element points to the data object in the dataObjectSection that this content unit describes. | dataObjectPointerType | 0 .. 1 |

## B2 Metadata Section

The manifest file contains exactly one metadata section. The metadata section contains a list of metadata objects that contain either wrapped metadata (information included directly in the manifest file), a data object pointer that refers to a physical auxiliary data file on disk, or a metadata reference that points to a representation data set schema file on disk.

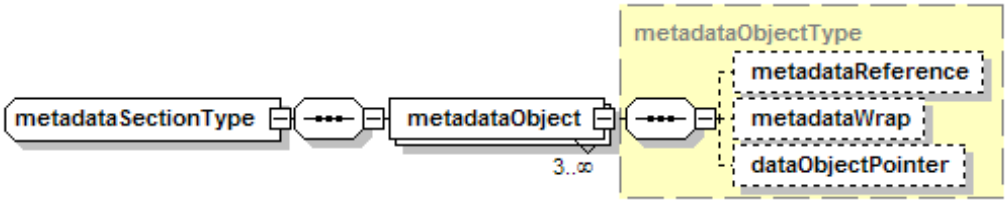


**Figure B-3 Metadata Section Structure and Content**

**Table B-4 Data Type - metadataSectionType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| metadataObject | Metadata objects can take one of three forms: the first is "wrapped metadata" in which valid XML data is embedded directly in the manifest file itself using an metadataWrap element; the second is a reference to the auxiliary data file in the dataObjectSection through a dataObjectPointer element; and, the third is a physical reference to a representation data set on the filesystem through the use of a metadataReference element. For Sentinel-1 auxiliary products a minimum of 3 metadataObjects shall be present dedicated to: processing, platform and generalProductInformation metadataWrap objects. Representation metadata objects shall also be provided if they apply/are available. | metadataObjectType | 3 .. ∞ |

**Table B-5 Data Type - metadataObjectType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *ID* | *Unique identifier of this meta data object.* | *ID* | *required* |
| *category* | *Defines the category of this meta data. The category is used to specify the nature of the metadata, whether it is preservation information (PDI), description information (DMD) or representation information (REP)* | *string* | *required* |
| *classification* | *A textual description of the classification of this meta data. The classification is linked to the category and provides a more verbose description of the nature of the metadata, whether it is preservation information (PROVENANCE), description information (DESCRIPTION) or representation information (REPRESENTATION)* | *string* | *required* |
| dataObjectPointer | The dataObjectPointer element is used when the metadata object is an auxiliary data file. The dataObjectPointer element is used to point to the applicable auxiliary data file in the dataObject section through its dataObjectID attribute. | dataObjectPointerType | 0 .. 1 |
| metadataWrap | The metadataWrap element is used to embed XML metadata directly in the manifest file itself. This element is used to express information that can be used for auxiliary product identification. | metadataWrapType | 0 .. 1 |
| metadataReference | The metadataReference element is used when the metadata object is a representation data set. The metadataReference element is used to specify the physical file location of the applicable representation data set. | metadataReferenceType | 0 .. 1 |

## B2.1 Mandatory Wrapped Metadata

The three mandatory wrapped metadata elements are presented in tabular form below.

**Table B-6 Mandatory Wrapped Metadata**

| Name | | | Description | Data Type | Cardinality |
|---|---|---|---|---|---|
| safe:processing | | | Metadata describing the processing steps performed on the auxiliary data. | processingType | 1 |
| | *name* | | *Name of the processing step used to create the auxiliary data.* | *string* | *required* |
| | *start* | | *Processing start time.* | *dateTime* | *required* |
| | *stop* | | *Processing stop time.* | *dateTime* | *required* |
| | safe:facility | | Identifies an organization authority of the processing step. | facilityType | 1 |
| | | *country* | *Name of the country where the facility is located.* | *string* | *required* |
| | | *name* | *Name of the facility where the processing step was performed.* | *string* | *required* |
| | | *organization* | *Name of the organization responsible for the facility.* | *string* | *required* |
| | | *site* | *Geographical location of the facility.* | *string* | *required* |
| | safe:software | | Reference to the software used for the processing step. | softwareType | 0 .. ∞ |
| | | *name* | *Name of the software.* | *string* | *required* |
| | | *version* | *Software version identification.* | *string* | *optional* |
| | safe:resource | | Reference to resources involved in the processing. | resourceType | 0 .. ∞ |
| | | *name* | *Name of the resource.* | *string* | *required* |
| | | *role* | *Role the resource played in processing.* | *string* | *required* |
| | | *href* | *URL of the resource.* | *anyURI* | *optional* |

| Name | | | Description | Data Type | Cardinality |
|---|---|---|---|---|---|
| safe:platform | | | Metadata describing the mission platform to which the auxiliary data applies. | platformType | 1 |
| | safe:nssdc Identifier | | Univocally identifies the mission according to standard defined by the World Data Center for Satellite Information (WDC-SI), available at http://nssdc.gsfc.nasa.gov/nmc/scquery.html | string | 1 |
| | safe:familyName | | The full mission name. E.g. “SENTINEL-1” | string | 1 |
| | safe:number | | The alphanumeric identifier of the platform within the mission. | string | 1 |
| | safe:instrument | | Information related to the instrument on the platform to which this auxiliary data applies. | instrumentType | 1 |
| | | safe:familyName | Instrument name. | string | 1 |
| | | *abbreviation* | *Abbreviated instrument name.* | *string* | *required* |
| s1auxsar:standAlone Product Information | | | Metadata describing the auxiliary product. | standAloneProduct InformationType | 1 |
| | s1auxsar: auxProductType | | Informative name of the type of this auxiliary product | auxProductTypeType | 0 .. 1 |
| | s1auxsar:validity | | Date and time at which the auxiliary product becomes valid. | dateTime | 1 |
| | s1auxsar:generation | | Date and time at which the auxiliary product was generated. | dateTime | 1 |
| | s1auxsar:instrument Configuration Id | | Identifier of the instrument (radar database) configuration to which this auxiliary data applies. | unsignedInt | 1 |
| | s1auxsar:changeDes cription | | Description of the changes of this auxiliary product with respect to the previous version | string | 0...1 |

## B3 Data Object Section

The manifest file contains exactly one data object section. The data object section contains a list of data objects that contain references to the physical auxiliary data files on disk.

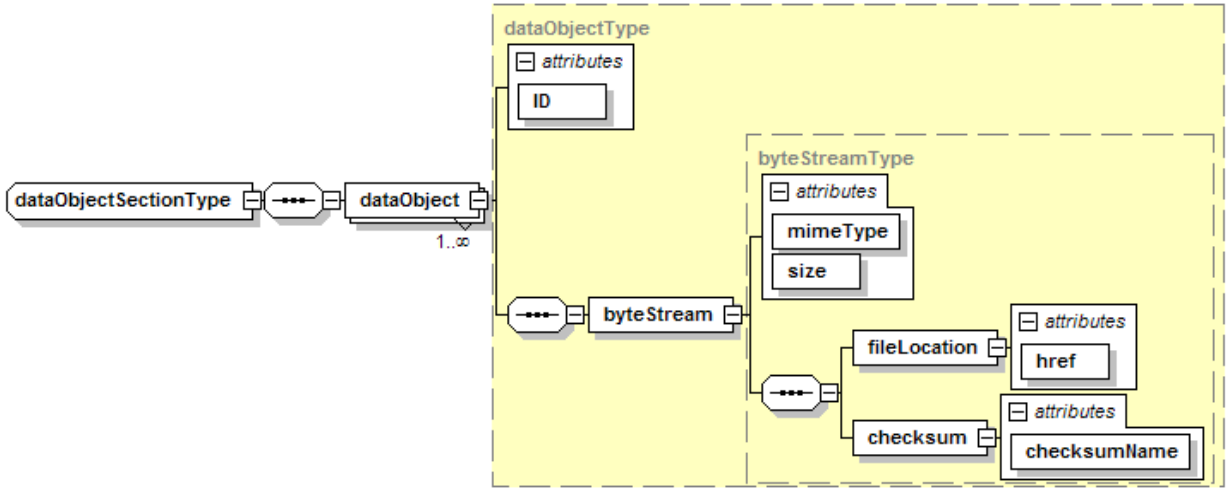


**Figure B-4 Data Object Section Structure and Content**

**Table B-7 Data Type - dataObjectSectionType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| dataObject | Each data object refers to a physical file on the filesystem through the use of its byteStream element. The mandatory ID attribute is used by elements in the informationPackageMap and the metadataSection to refer to these physical data objects. | dataObjectType | 1 |

**Table B-8 Data Type - dataObjectType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *ID* | *Unique identifier of this data object.* | *ID* | *required* |
| byteStream | The byte stream element points to the physical file that this data object represents. The byteStream element contains the location of the file and associated information like the format of the file, the size and the data integrity checksum. | byteStreamType | 1 |

**Table B-9 Data Type - byteStreamType**

| Name | Description | Data Type | Cardinality |
|---|---|---|---|
| *mimeType* | *Specifies the format of the file referred to by this byteStream element.* | *mimeType* | *required* |
| *size* | *Indicates the size (in bytes) of the file referred to by this byteStream element.* | *long* | *required* |
| fileLocation | The fileLocation element contains the absolute path or URL to associated file through the use of its “href” attribute. | referenceType | 1 |
| checksum | *Provides the integrity checksum for the file referred to by this byteStream element.* | checksumInformationType | 1 |