Antenna Database
AISG-ST-ADB
vADB3.1.1.6

Revision History

DATE	ISSUE	NOTES
5th November 2018	vADB3.1.1.6	First public relase.

Antenna Interface Standards Group
Subunit Type Standard AISG-ST-ADB vADB3.1.1.6

5th November 2018

1. FOREWORD 3
2. SCOPE 4
3. REFERENCES 5
4. VERSION COMPLIANCE 6
5. ABBREVIATIONS 7
6. TERMINOLOGY 8
7. DEFINITIONS 9
8. GENERAL ASPECTS 10
8.1. General 10
8.2. Subunit association 10
8.3. Array 10
8.4. Return codes 10
8.5. Resumption of operation 10
9. LAYER 1 11
9.1. DC power consumption 11
10. LAYER 2 12
11. LAYER 7 13
11.1. Subunit type 13
11.2. Overview of commands for ADB subunits 13
11.3. Bearing representation 13
11.4. Mechanical tilt representation 14
11.5. ADB commands 14
11.5.1. ADB Get Antenna Info 14
11.5.2. ADB Get Antenna Port Info 15
11.5.3. ADB Get Antenna Array Info 17
11.5.4. ADB Set Antenna Installation Info 19
11.5.5. ADB Get Antenna Installation Info 23
11.5.6. ADB Set RF Path ID to Array 24
11.5.7. ADB Get RF Path ID of Array 26
ANNEX A 28

1. FOREWORD

This standard has been produced by the Antenna Interface Standards Group (AISG) to introduce and define new features and enhancement of the management system for antenna line devices (ALDs) with remote control and monitoring facilities. AISG v3.0 base document describes the common behaviour of antenna line devices with AISG interfaces and type-specific functionality is defined in subunit type standards. This subunit type standard covers the antenna line devices capable of storing the antenna information.

For purposes of compliance and AISG interoperability, users should note that the implementation of this subunit type standard is optional. However, once it is selected for inclusion in a product, the entire standard becomes mandatory.

This standard is independent of previous 3GPP specifications.

2. SCOPE

AISG v3.0 specifies the standard data interface between a primary, typically a base station, and antenna line devices (ALDs) which are manageable units, usually associated with base station antennas.

The standard is divided into the base document and several subunit type standards. This subunit type standard document describes the specific behaviour of the Antenna Database (ADB) subunit type.

This standard defines the functional behaviour of ADB subunits. The text of the standard defines explicitly what is required or permitted. Anything that is not explicitly allowed is not permitted.

3. REFERENCES

This AISG Standard incorporates provisions from other publications. These are cited in the text and the referenced publications are listed below. Where references are listed with a specific version or release, subsequent amendments or revisions of these publications apply only when specifically incorporated by amendment or revision of this AISG standard. For references listed without a version or release, the latest edition of the publication referred to applies.

1 AISG v3.0: “AISG v3.0”

2 AISG v3.0 STCM: “Subunit Type Compliance Matrix”

3 AISG APCC: “Antenna port colour coding standard”

4 170217 NGMN P-BASTA Whitepaper v10.0: “Recommendation on Base Station Antenna Standards”

4. VERSION COMPLIANCE

The compliance of this standard with different version of AISG v3 baseline standard is defined in [2].

5. ABBREVIATIONS

Where abbreviations or acronyms are used in this document they have the following meanings:

ADB	Antenna Database
ALD	Antenna Line Device
ID	Identifier
LHC	Left Hand Circular (Polarisation)
MALD	Multi-primary ALD
NGMN	Next Generation Mobile Networks
RET	Remote Electrical Tilt
RF	Radio Frequency
RHC	Right Hand Circular (Polarisation)
SALD	Single-primary ALD
TCC	Time-Consuming Command
Xpol+	Cross Polarisation, slant +45 degrees
Xpol-	Cross Polarisation, slant -45 degrees
3GPP	3rd Generation Partnership Project

6. TERMINOLOGY

Where the following terms are used in this document, they have the following meanings:

Antenna port	RF port of the antenna with direction towards the basestation.
Array direction reference	The direction from which the antenna is viewed to define its polarisation. Possible values are front and back. (Some vendors determine the array direction by looking at the antenna from the front and others by looking at the antenna from the back).
Array polarisation axis reference	The axis from which the polarisation angle is measured in a clockwise direction. Possible values are horizontal or vertical. This assumes that the antenna is installed in its intended orientation.
Array	An array is a logical group of single or dual polarized radiators inside the antenna radome supporting a common frequency band and a common beam shape and tilt [3].
Mechanical bearing	The direction orthogonal to the axis of the antenna assembly, expressed in degrees East of the True North (ETN).
Mechanical tilt	Tilt angle of the antenna in the vertical plane. Tilt at an angle below straight and level shall be represented by a positive number (down-tilt), while tilt at an angle above straight and level shall be represented by a negative number. Tilt is reported in decimal degrees, to one decimal place of accuracy, and then multiplied by 10 so that is may be represented by an integer.
Polarisation	Orientation of the electric field vector of the radio wave emitted by an array.

7. DEFINITIONS

Provenance_t is used to identify the origin of the data.

```
Enumeration Provenance_t : uint8_t {
      NotSet          ← 0
      Factory         ← 1
      File            ← 2
      Automatic       ← 3
      Manual          ← 4
}
```

8. GENERAL ASPECTS

8.1. General

There shall exist only one ADB subunit for each antenna.

8.2. Subunit association

An ADB subunit may be associated with several ports without any interconnection between these ports.

8.3. Array

The definition of array means that if the two polarisations of a physical dual polarized array are controlled by one common RET subunit, they are considered to be one dual polarized array.

Furthermore, if the two polarisations of physical dual polarized array are controlled by two independent RET subunits, they are considered as two independent single polarized arrays.

One RET subunit can control any number of arrays.

8.4. Return codes

This subunit type standard introduces the following subunit type specific return codes.

```
Enumeration ReturnCode_t : uint16_t {
     ADBNotAntennaPort ← 0x0300                // e.g. a RET-port
}
```

8.5. Resumption of operation

The following data shall be retained after reset:

- Antenna Installation data
- RF Path to Array data

9. LAYER 1

All definitions and specifications for ALDs in [1] regarding layer 1 shall be valid for ALDs which contain ADB subunits.

9.1. DC power consumption

ADB is not allowed to switch to HighPowerMode.

10. LAYER 2

All definitions and specifications for ALDs in [1] regarding layer 2 shall be valid for ALDs which contain ADB subunits.

11. LAYER 7

An ALD which contains an ADB subunit shall support the command set which is defined in [1] in addition to those commands specified in this standard.

11.1. Subunit type

Subunit type	1-octet unsigned integer code
ADB	0x03

Table 11.1-1: Subunit type code

11.2. Overview of commands for ADB subunits

The table below shows an overview of all commands used in this ADB subunit type standard.

The following abbreviations are used in the Table 11.2-1: “Command set for ADB subunits”.

M Mandatory

O Optional

- Not applicable

ADB Command	Code	Initiator	Subunit	Timeout	TCC	Mandatory for:			Changes the ConnectionState	Changes to HighPowerMode	Mimimum required authority
						Primary	SALD	MALD			
ADB Get Antenna Info	0x0300	Primary	>0	1 s	no	M	M	M	no	no	RO
ADB Get Antenna Port Info	0x0301	Primary	>0	1 s	no	M	M	M	no	no	RO
ADB Get Antenna Array Info	0x0302	Primary	>0	1 s	no	M	M	M	no	no	RO
ADB Set Antenna Installation Info	0x0303	Primary	>0	1 s	no	M	M	M	no	no	RW
ADB Get Antenna Installation Info	0x0304	Primary	>0	1 s	no	M	M	M	no	no	RO
ADB Set RF Path ID to Array	0x0305	Primary	>0	1 s	no	O	M	M	no	no	RW
ADB Get RF Path ID of Array	0x0306	Primary	>0	1 s	no	O	M	M	no	no	RO

Table 11.2-1: Command set for ADB subunits

11.3. Bearing representation

The bearing range supported is 0.0° – +359.9° East of True North. The bearing value is expressed in 0.1° units with a range of 0 – +3599.

11.4. Mechanical tilt representation

The mechanical tilt range supported is −90.0 – +90.0. The bearing value is expressed in 0.1° units with a range of −900 – +900.

11.5. ADB commands

11.5.1. ADB Get Antenna Info

Description (Informative):

On the receipt of this command the ADB subunit returns antenna information.

The behaviour of the antenna model number and antenna serial number fields are vendor specific and it is possible that if a field replaceable submodule is exchanged in the field, info in these fields changes or is lost. The user may have to maintain this information manually during a field submodule replacement as it is not mandatory for the vendor to provide a mechanism for reading this information to the new submodule from the antenna.

Specification (Normative):

```
PrimaryCommand ADBGetAntennaInfoCommand {
      CommandCode_t         Command ← 0x0300
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength ← 0
}

ALDResponse ADBGetAntennaInfoResponse {
      CommandCode_t         Command ← 0x0300
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
            uint8_t         LengthOfAntennaModelNumber      // max 32 octets
            UTF8String_t    AntennaModelNumber
            Provenance_t    AntModelNumberProvenance
            uint8_t         LengthOfAntennaSerialNumber     // max 32 octets
            UTF8String_t    AntennaSerialNumber
            Provenance_t    AntSerialNumberProvenance

            uint16_t        NrOfArrays
            Provenance_t    NrOfArraysProvenance
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState

      }
}
```

```
Enumeration ReturnCode_t {
      OK
      InvalidSubunitNumber
      InvalidSubunitType
      UnknownCommand
      FormatError
      Busy
      IncorrectState
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF ALDType = MALD
AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] = NoAccess THEN
RETURN NotAuthorised
EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState,
RestrictedConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
RETURN result.code
EXIT
ENDIF

RETURN OK, AntennaModelNumber, AntennaSerialNumber, NrOfArrays and the corresponding lengths and provenances
CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.2. ADB Get Antenna Port Info

Description (Informative):

On the receipt of this command the ADB subunit returns antenna port information.

Specification (Normative):

```
PrimaryCommand ADBGetAntennaPortInfoCommand {
      CommandCode_t         Command ← 0x0301
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength ← 2
      uint16_t              PortNumber
}
```

```
ALDResponse ADBGetAntennaPortInfoResponse {
      CommandCode_t         Command ← 0x0301
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
            uint16_t NrOfArraysConnectedToThePort
            for(i = 0; i < NrOfArraysConnectedToThePort; i++)
            {
                  uint8_t         LengthOfArrayIDConnectedToThePort    // max. 6
                                                                       // octets
                  TextString_t    ArrayIDConnectedToThePort
                  Provenance_t    ArrayIDConnectedToThePortProvenance
            }
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState
      }
}

Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidPortNumber
      InvalidSubunitNumber
      InvalidSubunitType
      ADBNotAntennaPort
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF Cmd.PortNumber is not one of 1…MaxPort THEN
    RETURN InvalidPortNumber
    EXIT
ENDIF

IF ALDType = MALD
    AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] = NoAccess THEN
    RETURN NotAuthorised
    EXIT
ENDIF

IF (PortProperties[Cmd.PortNumber] bitwise AND RF) ≠ RF THEN
    RETURN ADBNotAntennaPort
    EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState,
RestrictedConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
    RETURN result.code
    EXIT

RETURN OK, number of ArrayIDs connected to the port, their array IDs and the corresponding lengths and provenances
CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.3. ADB Get Antenna Array Info

Description (Informative):

On the receipt of this command the ADB subunit returns antenna array information.

By combining the array direction reference, the polarisation axis reference and the polarisation value, all specified by the manufacturer and available through the ADB commands, the array polarisation is specified in a vendor independent and comparable way. This combination is only performed for slant linear polarisation.Array direction reference: Polarisation is defined looking at the antenna from the front or from the back [Value: Front or Back].

Polarisation angle reference: Starting from the vertical, to define polarisation is oriented 45 degrees clockwise or 45 degrees counter clockwise [Value: Clockwise or Counterclockwise].

For examples see Annex A

Specification (Normative):

```
Bitfield ArrayPolarisation_t : uint8_t {
      Xpol+      : Bit 0
      Xpol-      : Bit 1
      Vertical   : Bit 2
      Horizontal : Bit 3
      RHC        : Bit 4
      LHC        : Bit 5
}

Enumeration ArrayDirectionReference_t : uint8_t {
      Front ← 0            // As seen looking at the front of the
                           // antenna
      Back ← 1             // As seen looking at the back of the
                           // antenna
}

Enumeration ArrayPolarisationAxisReference_t : uint8_t {
      CW ← 0               // Polarisation reference axis is horizontal
      CCW ← 1              // Polarisation reference axis is vertical
}

PrimaryCommand ADBGetAntennaArrayInfoCommand {
      CommandCode_t        Command ← 0x0302
      CommandSequence_t    PrimaryCommandSequence
      Subunit_t            Subunit
      DataLength_t         DataLength ← 2
      uint16_t             ArrayNumber
}
```

```
ALDResponse ALDGetAntennaArrayInfoResponse {
      CommandCode_t         Command ← 0x0302
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
            uint16_t                  RelativeArrayPositionX
            Provenance_t              RelativeArrayPositionXProvenance
            uint16_t                  RelativeArrayPositionY
            Provenance_t              RelativeArrayPositionYProvenance
            uint16_t                  AzimuthBeamwidth3dB // see [4]
                                                          // paragraph 3.2.5
            Provenance_t              AzimuthBeamwidth3dBProvenance
            uint16_t                  Gain                // max value in dBi
            Provenance_t              GainProvenance
            Frequencies_t             ArrayFrequencies    // see Chapter 11 in
                                                          // AISG v3.0 base
                                                          // standard
            Provenance_t              ArrayFrequenciesProvenance
            ArrayPolarisation_t       ArrayPolarisation[3]
            Provenance_t              ArrayPolarisationProvenance
            PolarisationReference_t   ReferenceValue
            Provenance_t              ReferenceValueProvenance
            ArrayDirectionReference_t ArrayDirectionReference
            Provenance_t ArrayDirectionReferenceProvenance
            ArrayPolarisationAxisReference_t ArrayPolarisationAxisReference
            Provenance_t ArrayPolarisationAxisReferenceProvenance
      }
      else {
            ALDState_t          ALDState
            ConnectionState_t   ConnectionState
      }
}

Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidArrayNumber
      InvalidSubunitNumber
      InvalidSubunitType
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF Cmd.ArrayNumber is not one of 1..MaxArray THEN
    RETURN InvalidArrayNumber
    EXIT
ENDIF

IF ALDType = MALD
    AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] = NoAccess THEN
    RETURN NotAuthorised
    EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState,
RestrictedConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
RETURN result.code
EXIT
ENDIF

RETURN OK, Antenna array information and the corresponding provenances
CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.4. ADB Set Antenna Installation Info

Description (Informative):

This command provides a method by which installation related data can be written to the non-volatile memory in the ADB.

On the receipt of this command the ADB subunit stores installation data in non-volatile memory. The bitfield InstallationDataToBeWritten controls which data fields are stored. The same bit in the bitfield controls the storage of the length of the data (where applicable), the data itself and its provenance.

The bits in the bitfield controls the storage of data as follows:

Bit value 1: Corresponding data is stored together with length (where applicable) and provenance. Existing data is overwritten.

Bit value 0: No data is written to the non-volatile memory and existing data is preserved. Any data matching the bit present in the message is ignored.

Specification (Normative):

```
Bitfield DataToBeStored_t : uint8_t {
     SectorID             : Bit 0
     PositionWithinSector : Bit 1
     MechanicalBearing    : Bit 2
     MechanicalTiltValue  : Bit 3
}
```

```
PrimaryCommand ADBSetAntennaInstallationInfoCommand {
      CommandCode_t         Command ← 0x0303
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength
      DatatToBeStored_t     InstallationDatatoBeStored
      uint8_t               LengthOfSectorID                // max 32 octet
      TextString            SectorID
      Provenance_t          SectorIDProvenance
      uint8_t               LengthOfPositionWithinSector
      TextString_t          PositionWithinSector
      Provenance_t          PositionWithinSectorProvenance
      uint16_t              MechanicalBearing
      Provenance_t          MechanicalBearingProvenance
      uint16_t              MechanicalTilt                  // Mechanical
                                                            // tilt in
                                                            // degrees
      Provenance_t          MechanicalTiltProvenance
}

ALDResponse ADBSetAntennaInstallationInfoResponse {
      CommandCode_t         Command ← 0x0303
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState
            uint8_t               ParameterNumber
      }
}

Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidSubunitNumber
      InvalidSubunitTYpe
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF ALDType = MALD
AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] ≠ ReadWrite THEN
RETURN NotAuthorised
EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState,
RestrictedConnectionState},
Cmd.Command, CurrentPort)

```
UNLESS result.allowed THEN
    RETURN result.code
    EXIT
ENDIF

IF (Cmd.InstallationDataToBeStored.InstallersID) THEN
    IF Cmd.LengthOfInstallersID > 32 THEN
        Response.ParameterNumber ← 0
        RETURN OutOfRange
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
    ELSEIF Cmd.InstallersIDProvenance NOT IN (Manual, Automatic) THEN
        Response.ParameterNumber ← 0
        RETURN InvalidProvenance
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
    ELSE
        Store the LengthOfInstallersID, InstallersID and InstallersIDProvenance to non-
        volatile memory
    ENDIF
ENDIF

IF (Cmd.InstallationDataToBeStored.BaseStationID) THEN
    IF Cmd.LengthOfBaseStationID > 32 THEN
        Response.ParameterNumber ← 1
        RETURN OutOfRange
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
    ELSEIF Cmd.BaseStationIDProvenance NOT IN (Manual, Automatic) THEN
        Response.ParameterNumber ← 1
        RETURN InvalidProvenance
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
    ELSE
        Store the LengthOfBasestationID, BaseStationID and BaseStationIDProvenance
        to non-volatile memory
    ENDIF
ENDIF

IF (Cmd.InstallationDataToBeStored.SectorID) THEN
    IF Cmd.LengthOfSectorID > 32 THEN
        Response.ParameterNumber ← 2
        RETURN OutOfRange
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
    ELSEIF Cmd.SectorIDProvenance NOT IN (Manual, Automatic) THEN
        Response.ParameterNumber ← 2
        RETURN InvalidProvenance
        CommandExit(Cmd.Command, CurrentPort)
        EXIT
```

```
        ELSE
        Store the LengthOfSectorID, SectorID and SectorIDProvenance to non-volatile memory
        ENDIF
ENDIF

IF (Cmd.InstallationDataToBeStored.LengthOfPositionWithinSector) THEN
        IF Cmd.LengthOfPositionWithinSector > 32 THEN
                Response.ParameterNumber ← 3
                RETURN OutOfRange
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
        ELSEIF Cmd.PositionWithinSectorProvenance NOT IN (Manual, Automatic) THEN
                Response.ParameterNumber ← 3
                RETURN InvalidProvenance
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
        ELSE
                Store the LengthOfPositionWithinSector, PositionWithinSector and
                PositionWithinSectorProvenance to non-volatile memory
        ENDIF
ENDIF

IF (Cmd.InstallationDataToBeStored.MechanicalBearing) THEN
        IF Cmd.MechanicalBearing > 3599 THEN
                Response.ParameterNumber ← 4
                RETURN OutOfRange
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
        ELSEIF Cmd.MechanicalBearingProvenance NOT IN (Manual, Automatic) THEN
                Response.ParameterNumber ← 4
                RETURN InvalidProvenance
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
        ELSE
                Store the MechanicalBearing and MechanicalBearingProvenance to non-volatile
                memory
        ENDIF
ENDIF

IF (Cmd.InstallationDataToBeStored.MechanicaTiltValue) THEN
        IF Cmd.MechanicalTiltValue > 900 OT Cmd.MechanicalTiltValue < −900 THEN
                Response.ParameterNumber ← 5
                RETURN OutOfRange
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
        ELSEIF Cmd.MechanicalBearingProvenance NOT IN (Manual, Automatic) THEN
                Response.ParameterNumber ← 5
                RETURN InvalidProvenance
                CommandExit(Cmd.Command, CurrentPort)
                EXIT
```

ELSE
Store the MechanicalTiltValue and MechanicalTiltValueProvenance to non-volatile memory
ENDIF
ENDIF

IF the ALD detects a hardware error THEN
RAISE AlarmGeneralError Severity Major on Cmd.Subunit, “Hardware error”
RETURN GeneralError
ELSE
RETURN OK
ENDIF

CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.5. ADB Get Antenna Installation Info

Description (Informative):

On the receipt of this command the ADB subunit returns the installation data from the non-volatile memory.

Specification (Normative):

```
PrimaryCommand ADBGetAntennaInstallationInfoCommand {
      CommandCode_t         Command ← 0x0304
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength ← 0
}

ALDResponse ADBGetAntennaInstallationInfoResponse {
      CommandCode_t         Command ← 0x0304
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
            uint8_t               LengthOfSectorID                // max 32 octet
            TextString_t          SectorID
            Provenance_t          SectorIDProvenance
            uint8_t               LengthOfPositionWithinSector
            TextString_t          PositionWithinSector
            Provenance_t          PositionWithinSectorProvenance
            uint16_t              MechanicalBearing
            Provenance_t          MechanicalBearingProvenance
            uint16_t              MechanicalTiltValue             // Mechanical
                                                                  // tilt in
                                                                  // degrees
            Provenance            MechanicalTiltValueProvenance
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState
      }
}
```

```
Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidSubunitNumber
      InvalidArrayNumber
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF ALDType = MALD
AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] = NoAccess THEN
RETURN NotAuthorised
EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState
RestrictedConnectionState,
MALDConfigConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
RETURN result.code
EXIT
ENDIF

RETURN OK, SectorID, PositionWithinSector, MechanicalBearing, MechanicalTilt and the corresponding lengths and provenances
CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.6. ADB Set RF Path ID to Array

Description (Informative):

On the receipt of this command the ADB subunit assigns the list of RF Path IDs to the specified array.

Specification (Normative):

NOTE: After any antenna line configuration change, the mapping of the RF Path ID must be revalidated and possibly regenerated.

```
PrimaryCommand ADBSetRFPathIDtoArrayCommand {
      CommandCode_t         Command ← 0x0305
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength
      uint16_t              ArrayNumber
      uint8_t               NrOfRFPathIDs
      for(i = 0; i < NrOfRFPathIDs; i++)
      {
            uint16_t   RFPathID
      }
}

ALDResponse ADBSetRFPathIDToArrayResponse {
      CommandCode_t         Command ← 0x0305
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState
      }
}

Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidSubunitNumber
      InvalidSubunitType
      InvalidArrayNumber
      TooManyArguments
      GeneralError
}
```

Primary specification (Normative):

ALD specification (Normative):

IF ALDType = MALD
    AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] ≠ ReadWrite THEN
    RETURN NotAuthorised
    EXIT
ENDIF

IF Cmd.ArrayNumber is not one of 1..MaxArray THEN
    RETURN InvalidArrayNumber
    EXIT
ENDIF

IF Cmd.NrOfRFPathIDs > 6 THEN
    RETURN TooManyArguments
    EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
RETURN result.code
EXIT
ENDIF

Store the RF path IDs for the supplied ArrayNumber to non-volatile memory

IF the ALD detects a hardware error THEN
RAISE AlarmGeneralError Severity Major on Cmd.Subunit, “Hardware error”
RETURN GeneralError
ELSE
RETURN OK
ENDIF

CommandExit(Cmd.Command, CurrentPort)
EXIT

11.5.7. ADB Get RF Path ID of Array

Description (Informative):

On the receipt of this command the ADB subunit returns the RFPathID list for the requested array number.

Specification (Normative):

```
PrimaryCommand ADBGetRFPathIDOfArrayCommand {
      CommandCode_t         Command ← 0x0306
      CommandSequence_t     PrimaryCommandSequence
      Subunit_t             Subunit
      DataLength_t          DataLength ← 2
      uint16_t              ArrayNumber
}

ALDResponse ADBGetRFPathIDOfArrayResponse {
      CommandCode_t         Command ← 0x0306
      CommandSequence_t     PrimaryCommandSequence
      ReturnCode_t          ReturnCode
      DataLength_t          DataLength
      if (ReturnCode == OK) {
            uint8_t NrOfRFPathIDs
            for(i = 0; i < NrOfRFPathIDs; i++){
                  uint16_t RFPathID
            }
      }
      else {
            ALDState_t            ALDState
            ConnectionState_t     ConnectionState
      }
}
```

```
Enumeration ReturnCode_t {
      OK
      FormatError
      Busy
      UnknownCommand
      IncorrectState
      InvalidSubunitNumber
      InvalidSubunitType
      InvalidArrayNumber
      NotAuthorised
}
```

Primary specification (Normative):

ALD specification (Normative):

IF ALDType = MALD
AND ActiveAuth[CurrentPort].Authority[Cmd.Subunit] = NoAccess THEN
RETURN NotAuthorised
EXIT
ENDIF

IF Cmd.ArrayNumber is not one of 1..MaxArray THEN
RETURN InvalidArrayNumber
EXIT
ENDIF

result ← IsCommandAllowed(LIST{ OperatingConnectionState
RestrictedConnectionState,
MALDConfigConnectionState},
Cmd.Command, CurrentPort)

UNLESS result.allowed THEN
RETURN result.code
EXIT
ENDIF

RETURN OK, number of stored RF path IDs and the list of RF path IDs of the requested Array
CommandExit(Cmd.Command, CurrentPort)
EXIT

ANNEX A

Examples and clarifying diagrams will be added in the next release of this document.